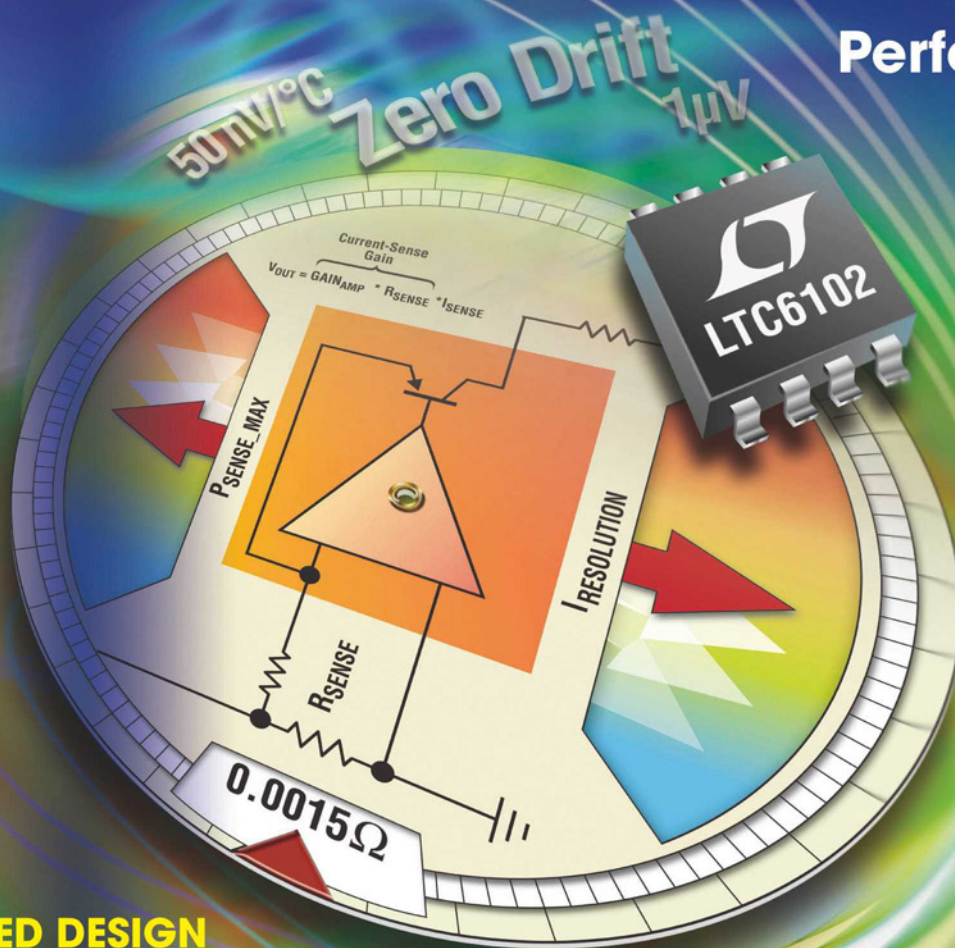


Electronics WORLD

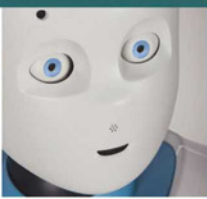
THE ESSENTIAL ELECTRONICS ENGINEERING MAGAZINE

**Precision
Amplifiers
Enable High
Performance
Current
Sensing**



EMBEDDED DESIGN SPECIAL REPORT:

- Programmable logic
- IoT wireless connectivity
- SOC design



Technology
The robot – your next
best friend



Smart Apps
Formula 1 in your
pocket



NEW Series
LTE design

THE WORLD'S LARGEST SELECTION OF ELECTRONIC COMPONENTS AVAILABLE FOR IMMEDIATE SHIPMENT!™



**OPEN
ACCOUNTS
AVAILABLE**
FOR QUALIFYING
CUSTOMERS



**LIVE
WEB CHAT**
*24/7, 365 DAYS
PER YEAR*



**FREE
SHIPPING
ON ORDERS
OVER £50***



**PRICES ARE IN
BRITISH POUND
STERLING AND
INCLUDE DUTIES**

**LOCAL
SALES &
TECHNICAL
SUPPORT
AVAILABLE**



0800 587 0991 • 0800 904 7786
DIGIKEY.CO.UK

1,000,000+ PRODUCTS IN STOCK | 650+ INDUSTRY-LEADING SUPPLIERS | 3.9 MILLION PARTS ONLINE

*A shipping charge of £12.00 will be billed on all orders of less than £50.00. All orders are shipped via UPS for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling and include duties. If excessive weight or unique circumstances require deviation from this charge, customers will be contacted prior to shipping order. Digi-Key is an authorized distributor for all supplier partners. New product added daily. © 2015 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ecia
MEMBER

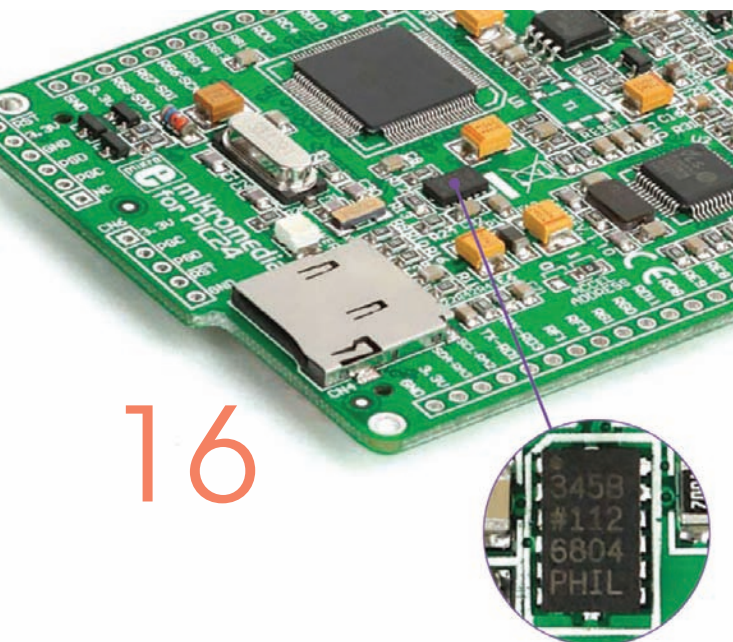
ecsn
member

CEDA
PARTNER

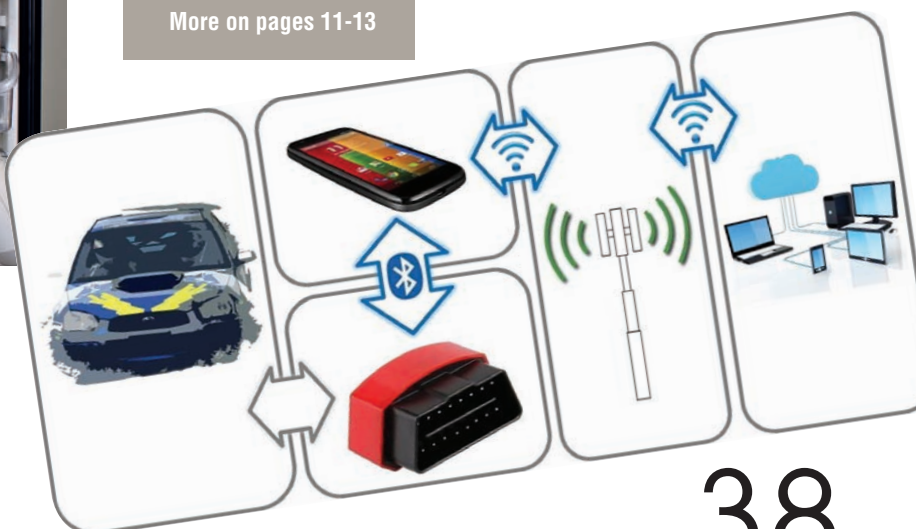


REGULARS

- 05 TREND**
KEEPING UP WITH THE FAST PACE OF CHANGE IN
AUTOMOTIVE ELECTRONICS
- 06 TECHNOLOGY**
- 08 RASPBERRY PI COLUMN**
- 14 LTE COLUMN**
- 16 EMBEDDED USER INTERFACE DESIGN
ON A BUDGET**
by Lucio Di Jasio
- 20 THE TROUBLE WITH RF...**
IDLE STATE
by Myk Dormer
- 37 INDUSTRY ANNOUNCEMENTS**
- 42 PRODUCTS**



Cover supplied by
LINEAR TECHNOLOGY
More on pages 11-13



38

FEATURES

- 22 EMBEDDED AND APPLICATION DEVELOPERS –
LEARNING FROM EACH OTHER**
By Greg Law, founder and CEO, Undo Software
- 26 SYSTEM DESIGN EXPLORATION WITH OPENCL
FOR FPGAS**
Dmitry Denisenko and Mykhailo Popryaga describe the design
of a two-million point frequency domain filter using Altera's
OpenCL software development kit for FPGAs
- 30 OPTIONS FOR PROVIDING IOT WIRELESS
CONNECTIVITY**
By Rui Ramalho, Product Manager for connectivity modules at
Murata Europe
- 32 TIME TO TAKE THE HEAT OUT OF LEDS**
Using ARM A9 cores on a Xilinx's Zynq system-on-a-chip can
significantly increase the performance of a system, writes
Adam P. Taylor, Chief Engineer for electrical systems at e2v
- 38 FORMULA 1 IN YOUR POCKET: TELEMTRY
FOR EVERYONE?**
By Tony Dewhurst, TrackSpy Development Lead

Disclaimer: We work hard to ensure that the information presented in Electronics World is accurate. However, the publisher will not take responsibility for any injury or loss of earnings that may result from applying information presented in the magazine. It is your responsibility to familiarise yourself with the laws relating to dealing with your customers and suppliers, and with safety practices relating to working with electrical/electronic circuitry – particularly as regards electric shock, fire hazards and explosions.

Switching for Every Application

***Because in Electronic Test,
One Platform Does Not Fit All!***



Pickering offers the largest range of switching products and platforms for all automated electronic Test & Measurement applications:

- The most switching choices in PXI, LXI and PCI formats
- Switching from low level DC to 1kV and RF/Microwave to 65GHz
- Pickering is the only switching vendor with **eBIRST™** Switching System Test Tools, which quickly and accurately locate switching system faults in all of their switching platforms. Visit **pickeringtest.com/ebirst** to learn more.



Pickering support all of their products with a standard three year warranty.

For more information scan this QR Code or go to **pickeringtest.com/advantage**



Switching | Instrumentation | Programmable Resistors | Custom Design | Cables & Connectors

pickeringtest.com     

Tel: +44 (0)1255-687900 | e-mail: sales@pickeringtest.com

Direct Sales Offices in USA, UK, Sweden, Germany, Czech Republic, France and China



KEEPING UP WITH THE FAST PACE OF CHANGE IN AUTOMOTIVE ELECTRONICS



The automotive sector has long pushed the boundaries in electronic design, pioneering applications that make their way into other industries over time. Considering the vast changes taking place in the automotive industry right now, the automotive electronics market is collectively in the middle of one of the most exciting times for decades: electric vehicles, collision avoidance systems, alternative fuel sources, driverless cars... the list goes on.

The automotive industry has some good reasons for this pace of change: the need to be competitive in a marketplace where differentiation is hard; demand for reduced production costs; faster time-to-market; the rigours of industry compliance such as with ISO 26262; and, of course, increasingly discerning consumers who expect more for their money. So it is no surprise that we are seeing automotive electronic systems that until a few years ago would have been reserved purely for the highest premium brands, with processing power found only in "super computers".

In turn, all these changes and demands put pressure on the electronics design team, which is why more companies are re-evaluating their tools and processes to ensure they are fit for purpose and able to keep up with the ever-evolving environment. Companies that are not reviewing their systems will be left behind by those modernising to enable the latest development best practice.

Agile development is not new and has become far more widespread than purely for software development. As organisations have awakened to this new way of working, where iterations and releases happen faster, they have realised the benefits of flexibility in their plans, enabling better alignment with their business needs. Similarly, continuous integration (CI) has gained popularity, involving building and testing embedded software products on a more regular basis, and if something doesn't work properly fixing it earlier and more cheaply.

A natural evolution from this is continuous delivery (CD) that, according to Evans Data research carried out in the first half of 2014, was then being adopted by over 60% of organisations surveyed in the UK and the US.

CD involves delivering products as soon as they are releasable, with the intent of getting early customer feedback to influence the next iteration. These shorter cycles enhance the alignment between development and customer, increase competitiveness and reduce risk by making small changes frequently. Such an incremental approach to development can apply across all elements of a project, not just code and other software components, but encompassing any asset in a project, for instance CAD material, web content, marketing documentation and other data associated with bringing a project to market.

Some might not consider cars to be objects that lend themselves to continuous delivery. Although it's true that for many physical components, various prototyping and manufacturing steps enforce delays or manual steps between developers and customers, the CD approach can still be adopted if delivery into the QA or pre-production stages is considered to be a "release".

With software becoming a prevalent component of modern vehicles, there are a variety of systems that lend themselves to incremental or regular updates which can be automated. For example, over-the-air (OTA) releases could be published to update infotainment systems with a new music-streaming service or new map data. Some manufacturers have begun to deploy OTA updates for more critical systems such as engine management.

These new methodologies may have a great deal to offer, but it does not necessarily follow that the legacy systems and tools will support them. Many automotive design and programming teams are using big, heavy and inflexible tools that began life as general manufacturing systems more suitable for numerically-controlled cutters than modern software development. Tools that haven't had a major change in the past five years are probably not going to be flexible enough, or sufficiently up-to-date, to cope with today's electronic software design environments. For example, the theories behind both agile and CD depend on transparency and visibility across all aspects of a project, from previous versions through to the current working model. This means that a single repository or 'source of truth' for all assets, not just software source code, is essential, with the added bonus of simplifying compliance management and reducing software maintenance costs.

Automation is also a major tenet of CD (and also of DevOps, which readers may have heard of in the context of more mainstream software development environments). Manual and repetitive tasks have the potential for error to creep in, and modern automation tools for software builds, testing and release management reduce these risks. A good example is the automation of compliance testing and simulators, which helps reduce the need for physical prototypes until the last possible moment, hence reducing cost and potential wasted time.

At the heart of many agile and CD projects sit version control systems, also known as version management. Traditional software developers may know them from back in the day as source code or SCM (software configuration management) systems, but the modern generation of these tools is designed to support every single asset within a project, if necessary across non-technical environments too. In fact, Perforce's own research a couple of years ago found that many of its users deploy version control for tasks other than managing software development.

One of the challenges of the automotive sector is the wide range of software that needs support: embedded software, hardware software, mobile apps, simulation models, web services and communications. Plus, designers work with a wide range of platforms, such as Java and .Net, and a variety of different development tools. As end products will be in the marketplace for some years, they need to be supported, including incremental feature releases, not just new developments. ●

By Mark Warren, Perforce Software (www.perforce.com)

EDITOR: Svetlana Josifovska
Tel: +44 (0)1732 883392
Email: svetlanaj@sjpbusinessmedia.com

SALES: James Magnani
Tel: +44 (0)20 7933 8976
Email: jamesm@sjpbusinessmedia.com

DESIGN: Tania King
PUBLISHER: Justyn Gidley

ISSN: 1365-4675
PRINTER: Buxton Press Ltd

SUBSCRIPTIONS:
Subscription rates:
1 year: £62 (UK);
£89 (worldwide)
Tel/Fax +44 (0)1635 879361/868594
Email: electronicsworld@circdata.com

SJP
business media

2nd Floor,
52-54 Gracechurch Street,
London, EC3A 0EH



Follow us on Twitter
@electrow



Join us on LinkedIn



NEW FD-SOI PROGRAM HELPS DESIGNERS CREATE LOW-POWER APPLICATIONS

Grenoble-based CEA-Leti has been joined by seven partners in support of its new FD-SOI (fully depleted silicon on insulator) IC development program, called Silicon Impulse. The initiative aims to provide a comprehensive technology platform for IC design, advanced intellectual property (IP) creation, emulation and test services, along with industrial multi-project wafer (MPW) shuttles.

The partners include CEI-Leti, CEI-List, STMicroelectronics, Dolphin Integration, CMP, Mentor Graphics, Cortus and Presto Engineering, who will combine their expertise and facilities for one-stop-shop services to speed development of energy-efficient products for ultra-low-power (ULP) Internet of Things (IoT), energy-efficient computing systems, and robust and reliable applications for harsh environments. The partners are also in discussion with Synopsys to join the program.

"Leti has always concentrated on research that helps our partners adopt technology to become more competitive in their markets," said Marie-Noëlle Semeria, CEO of Leti.

With the program's flexible format, Silicon Impulse's involvement can be limited to consulting or extended to developing and delivering the whole system, or anything in between. It can help innovators with their projects from concept through production hand-off. Companies can receive advice and have their products shaped from a very high level, including

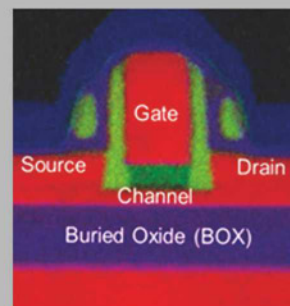
FD-SOI

Fully Depleted Silicon On Insulator, or FD-SOI, is a planar process technology that relies on two primary innovations.

First, an ultra-thin layer of insulator, called the buried oxide, is positioned on top of the base silicon. Then, a very thin silicon film implements the transistor channel. Thanks to its thinness, there is no need to dope the channel, thus making the transistor fully depleted.

The combination of these two innovations is called "ultra-thin body and buried oxide Fully Depleted SOI" or UTBB-FD-SOI.

FD-SOI enables much better transistor electrostatic characteristics versus conventional bulk technology. In addition, the buried oxide layer lowers the parasitic capacitance between source and drain and efficiently confines the electrons flowing from source to drain, dramatically reducing performance-degrading leakage currents.



a feasibility study and recommendations on how to implement the system. Leti and its partners can also provide unique IP and/or technology components such as foundation IP or more complex system-level IP blocks, RF, NVM, N/MEMS, 3D components and any other advanced technology to shape a unique and advanced, yet manufacturable, product. At another level, Leti and List can provide embedded software to complete the whole product.

MPW shuttles are offered to open the doors to a wider set of users and projects, enabling innovators to

test their ideas, especially mixed-signal, analog or RF technologies or any new IP that would require silicon validation in FD-SOI. This also provides an affordable platform for startups and other small companies to build their prototypes and run small volumes until they receive financing and/or demonstrate market traction to build their own mask set.

The first 28nm FD-SOI MPW is planned for February 2016 to be processed at STMicroelectronics's site in Crolles, near Grenoble.

THE WORLD'S FIRST EMOTIONS-READING PERSONAL ROBOT GOES ON SALE IN JAPAN

The world's first personal robot which reads emotions, named Pepper, went on sale in Japan in June. Some 1,000 units were available for



purchase immediately, with followup shipments scheduled from July onwards.

Pepper not only reads emotions, it has been created to have emotions, enabled by functions developed by cocoro SB. These emotion functions have been modelled on the release of hormones in humans in response to stimuli to the five senses, which in turn generate emotions.

Pepper's emotions are generated autonomously following processing of information from its cameras, touch sensors, accelerometers and other sensors within its "endocrine-type multi-layer neural network". Its emotions are influenced by people's

facial expressions and words, as well as its surroundings, which in turn affect its words and actions. For example, Pepper is at ease when it is around people it knows, happy when praised and scared when the lights go off.

Depending on the emotion at the time, Pepper raises or lowers its voice, sighs, exclaims, cries, and so on. Pepper's emotions can be seen on the heart display, which shows different colours and movements.

SoftBank, the company behind Pepper, plans to expand the lineup of robots and apps, including a dedicated model for businesses, called 'Pepper for Biz'.

ECOPAC ^{UK} Power _{LTD}



**Power
Supplies**

DIN Rail

Open Frame

Enclosed

External



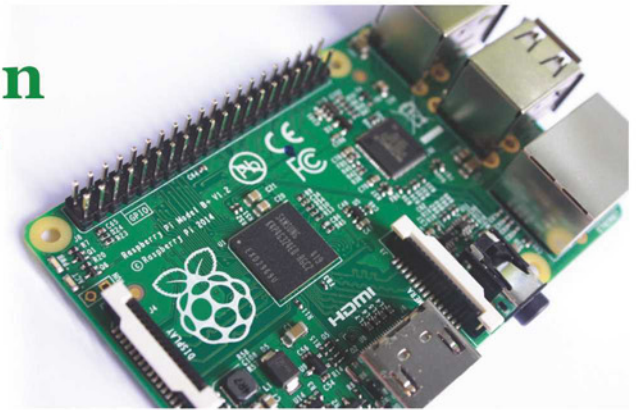
Call: 01844 204420

Email: sales@ecopacpower.co.uk

Website: ecopacpower.co.uk

THIS SERIES PRESENTS THE RASPBERRY PI SINGLE-BOARD COMPUTER, ITS FEATURES AND BENEFITS, AND ITS USE IN VARIOUS PROJECTS

Creating A Motion Sensor and Door Switch With The Raspberry Pi



BY **ANDREW ROBSON AND MIKE COOK**

In our first home-automation project we will create a motion sensor and door switch.

Motion sensors and door switches allow us not only to create alarms or alerts, but also to monitor flow and movement throughout a home. Using the information created by these sensors, you can apply rules and actions using Python. For example, you can turn off a light in a room if motion is not detected for a length of time, or receive an e-mail alert if your garage door is open for a period of time. So by combining sensors, rules and actions, you can create a more intelligent home environment.

A door switch contains a reed switch (see Figure 1), triggered when the two pieces come apart. One piece is attached to the door and the other to the door frame. One half contains a magnet and the other a reed switch held closed when the magnet is near. When the two come apart the magnetic field loses its strength and the switch is broken. When the door is closed, current flows through the switch, and when the door is open the current is broken. This type of switch is called normally open.

The motion sensor, sometimes known as a passive infrared (PIR) sensor (see Figure 2), is packaged with an integrated circuit and a lens that will focus the moving objects for the sensor. Both sensors interact with the Raspberry Pi through the GPIO port in the same manner, which makes their software identical.

You will need to write a Python script to monitor the sensors and produce an action when either sensor is triggered. In this

tutorial you will learn to trigger a message to the screen, but also take any number of other actions as explained before.

Project Parts

The following parts are needed for this project:

- 1 x door switch; easily obtained from a home security outlet or online.
- 1 x motion sensor; usually 5-12V with three terminals, +V_E, GND and 3.3V Pi-friendly output.
- 2 x 10kΩ resistors: brown, black, red, gold.
- 2 x 1kΩ resistors: brown, black, orange, gold.
- 1 x solderless breadboard; a prototyping board to which parts and wires can be connected by clipping them on to the board. It is used for prototyping electronics without having to solder parts together.
- Jumper wires: three male-to-male for breadboard connections, five male-to-female for connecting the breadboard to the GPIO pins. Jumper wires usually come in packs of various quantities, colours and sizes. Although only eight are needed for this project, having 20-30 of each are sufficient for most projects. Any size will do for this project, but shorter male-to-male (10cm) and longer male-to-female (20cm) are best.

Construction

The Raspberry Pi's GPIO ports are used to interface with the sensors, configured for input or output. In this case we will use them for input as data is collected from an external sensor.

A GPIO port can have three different states: high (positive), low (ground) or floating (either). To accurately set the state of the GPIO pin, tie it either to positive or negative (ground), by using a pull-up or pull-down resistor. When the GPIO is connected to a positive terminal, the GPIO pin's value is high; when connected to ground it is low.

Figure 3 shows a 10kΩ pull-up resistor that connects the GPIO to the positive terminal when the switch is open.

Figure 1: Door switch

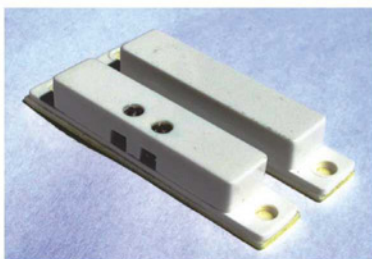
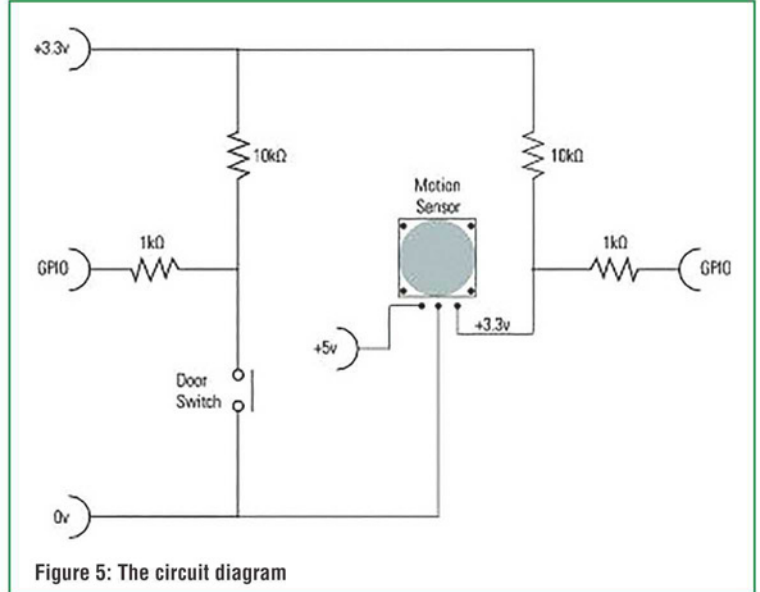
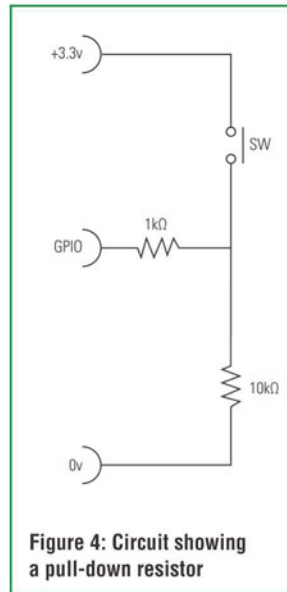
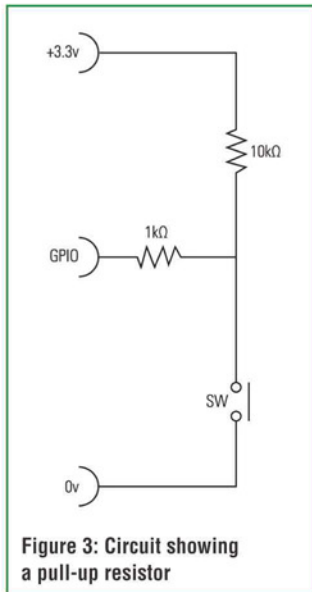


Figure 2: Motion sensor





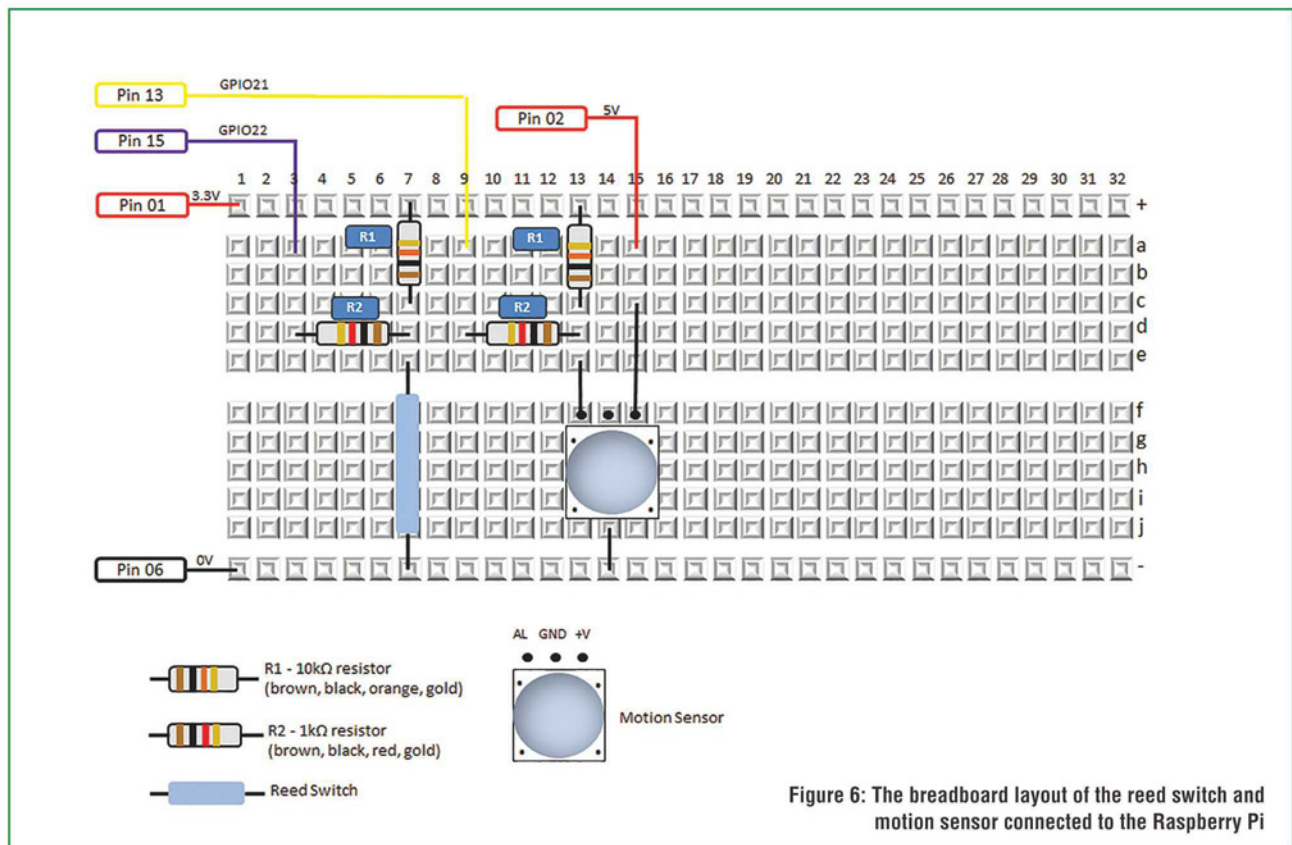
However, when the switch closes, there is a lower resistance path to ground and the GPIO will go low.

Figure 4 shows a pull-down resistor ($10k\Omega$) connecting the GPIO pin to ground, making it low. When the switch is pressed there is a lower resistance to positive, which changes the GPIO state to high. The $1k\Omega$ resistor is there to

protect the GPIO pin from a short circuit when the switch is pressed.

Tip

Always double-check the wiring before powering on the Raspberry Pi, as it can be very easily damaged by not



protecting the GPIO pins with resistors. Be especially careful of the 5V pins, which if connected to a GPIO pin will cause permanent damage.

In your project you want a closed circuit (low state) when the door is closed or when there is no motion, and a high state when the door is opened or motion is detected, so a pull-up resistor is needed to set the GPIO high in an alarm state – see the circuit diagram in Figure 3.

The PIR sensor has three coloured wires for ground, positive and alarm. The colours and pin positions may differ, depending on which model you have, so be sure to check its data sheet. The alarm pin is open-collector, meaning it requires pull-up resistor.

Software

The software will print to screen whenever motion is detected or the door sensor is triggered. Ideally, print to screen only once per event, and use variables (motion and door) to keep track of when the system is in alarm state and so prevent the trigger recurring for the same event. If logging these events to a database, or taking another action such as switching on a light or sending an e-mail alert, this method prevents multiple actions being created for the same event. The code is provided in Listing 1. ●

```
#!/usr/bin/env python
```

```
"""
```

Home Automation using motion detection
For the Raspberry Pi

```
"""
```

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(13, GPIO.IN)
GPIO.setup(15, GPIO.IN)
def main():
    motion = False
    door = False
    while True:
```

```
    if GPIO.input(13):
    if motion == False: print "Motion Detected"
    motion = True
    else:
    motion = False
    if GPIO.input(15):
    if door == False: print "Door Opened"
    door = True
    else:
    door = False
    time.sleep(.3)
    if __name__ == "__main__":
    main()
```

Listing 1: Home automation with motion detection

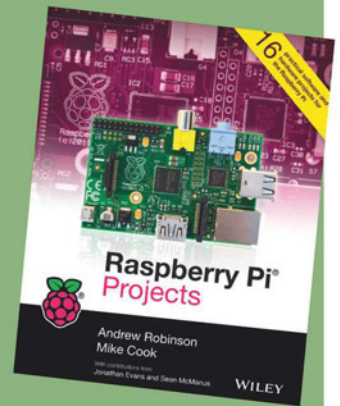
RASPBERRY PI PROJECTS

Raspberry Pi represents a new generation of computers that encourage the user to play and to learn and this book is aimed at the beginner Raspberry Pi user who is eager to get started creating real-world projects.

Containing 16 practical projects, this fun and informative resource introduces readers to the skills required in order to make the most of the Pi.

Raspberry Pi Projects will be available in paperback and e-book priced £14.99.

We have several copies of this book to give away. To register your interest, write to the Editor at svetlanaj@sjbusinessmedia.com



Electronics

WORLD

Your essential electronics engineering magazine and technical how-to-guide

12 monthly issues in digital format

Regular topical supplements

The Annual T&M Supplement

Weekly email bulletin

Comment and analysis from industry professionals

Tips and tricks

News and developments

Product reviews

Free invitations to our webinars

A subscription to Electronics World offers

Electronics World provides technical features on the most important industry areas, including:

Nano measurement

RF

Power supplies

DSPs

Automotive

Signal processing

Lighting

Connectors

Power supplies

Test and measurement

Cables

Embedded

Microwave

Robotics

Communications

Semiconductors

USB design

SUBSCRIBE TODAY FROM JUST £46 BY VISITING THE WEBSITE OR CALLING +44(0)1635 879 361

www.electronicsworld.co.uk/subscribe

Register for our free newsletter, please scan here

PRECISION AMPLIFIERS ENABLE HIGH PERFORMANCE CURRENT SENSING

By Greg Zimmer, Senior Product Marketing Engineer, Signal Conditioning, Linear Technology Corporation

M

ost analog ICs (comparators, op-amps, instrumentation amplifiers, references, filters, etc) are designed to handle voltage signals. When it comes to handling current signals, designers are left with a lot fewer options and a lot more headaches. That's unfortunate, since there can be a big advantage to directly monitoring and measuring current. Motor torque, solenoid force, LED intensity, solar cell exposure and battery power can be monitored best by looking at the current flow. What's needed is a circuit that can precisely sense current and convert this current into a voltage so it can be amplified, conditioned and measured by the readily available voltage devices (amplifiers, comparators, ADCs, etc). Although a resistor can translate current to voltage, a resistor by itself does not provide a complete solution. The most common solution is to use a sense resistor, placed directly in series with the current, and an amplifier to isolate and condition the voltage across this resistor (V_{SENSE}).

COMBINING AN AMPLIFIER & A SENSE RESISTOR

At first glance, placing a resistor in series with a ground contact may seem like the most straightforward current sensing approach. This technique, known as low-side current sensing (Figure 1a), requires that no ground paths exist that could allow current to be diverted around the sense resistor or that could contribute current from an adjacent circuit. If a mechanical frame establishes the system ground, it may be impractical to insert this sense resistor. Also, since grounds are not perfect conductors, ground voltage can vary at different points in the system, necessitating the use of a differential amplifier for precision measurements (Figure 1b).

There is a more serious problem when implementing low-side current sensing. A resistor in the ground path means that the "ground" of the load will change as the current changes. This can induce common mode errors in the system and presents a problem for interfacing to other systems requiring the same ground potential. Since measurement resolution is enhanced by the magnitude of V_{SENSE} , the designer must trade-off "ground noise" for increased resolution. A modest full-scale V_{SENSE} of

100mV translates to 100mV of injected ground noise. The problem of ground variation can be avoided by placing the current sense resistor between the power supply and the load.

This alternative approach is referred to as high-side current sensing. Again, the differential voltage across the sense resistor provides a direct measurement of current, however, now there is a non-zero common mode voltage across the resistor. This configuration presents the technical challenge that a small differential sense voltage must be discerned from the common mode voltage of the power supply (Figure 2).

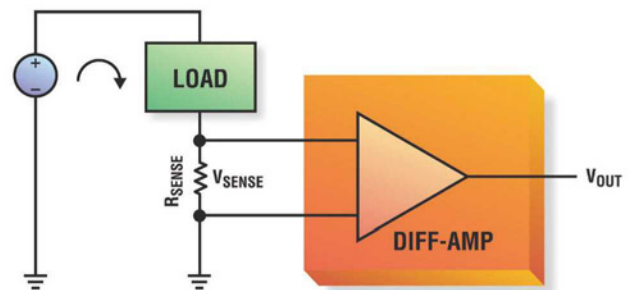


Figure 1a: Low-Side Current Sense Topology

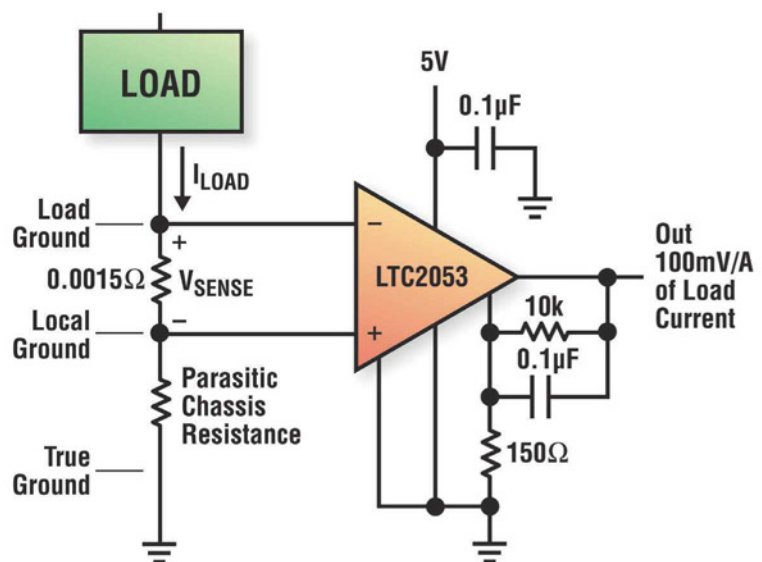


Figure 1b: Low-Side Current Sense Circuit Implemented

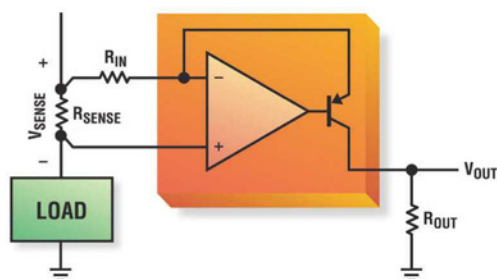


Figure 2: Basic High Side Current Sense Amplifier

For low voltage systems, an instrumentation amplifier or other rail-to-rail differential amplifier may suffice for monitoring a high-side sense resistor. The output of the amplifier must then be translated to ground without adding significant error. When the supply voltage is very high, circuitry may be required to translate V_{SENSE} down to the input common mode range of the amplifier or to float the amplifier up to the supply voltage. Aside from the added board space and cost, these techniques assume that the common mode voltage will stay within a narrow, specified range. For most current sense applications, it's very useful to anticipate large common mode fluctuations. For example, if the current sense circuit can operate when the power supply voltage drops, it can indicate if there is a problem at the supply or at the load; excessive current

suggests current limiting and load faults, and insufficient current indicates power supply failure. On the other hand, current sense circuits may face common mode voltages that exceed the supply voltage. Many current-devices, such as motors and solenoids, are inductive by nature; rapid current changes through them will cause inductive flyback, leading to large voltage swings across the sense resistor. It is precisely in these instances when the amplifier can be most useful.

SIMPLE SOLUTION

To address the challenges of current sensing, high-side current sense amplifiers were created. These special amplifiers are designed to extract a small differential voltage, generated by current passing through a small sense resistor, from a high common mode voltage. The sense voltage is then amplified and translated into a ground-referenced signal. Figure 2 illustrates the basic topology of a high side current sense amplifier. In this case, the amplifier forces a voltage across R_{IN} that is equivalent to V_{SENSE} . The current through R_{IN} is then forced through R_{OUT} , providing a ground referenced output voltage. For this basic capability, it is clear that high-side current sense amplifiers should have high input impedance, high gain with good gain accuracy, and a wide common mode range with good common mode rejection. What may not be as clear is the importance of the amplifier's precision.

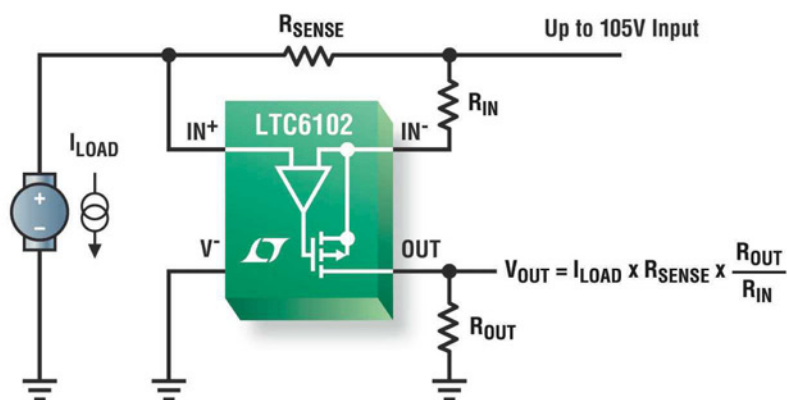


Figure 3: Linear Technology's LTC6102 provides a straightforward implementation of high-side current sensing. An R_{SENSE} and two gain resistors configure the part. The designer can customize power consumption, response time and the input/output impedance characteristics by their selection of R_{IN} and R_{OUT} .

FOCUS ON THE RESISTANCE

Ideally, the task of current and voltage sensing should not impact the load to which it's connected. This means that voltage sense devices should have nearly infinite input impedance; this ensures that no appreciable current can be diverted from the load. Conversely, a current sense device should have nearly zero input impedance; this ensures that the voltage to the load is not significantly reduced. High-side current sense circuits (amplifier + resistor) are subject to both of these requirements. The amplifier used to sense the voltage across R_{SENSE} must have high input impedance. The resistor used to sense the current to the load must be very small.

To fully appreciate this, let's consider the use of a large sense resistor. As the series resistance is increased, the voltage

available to the load is reduced. Extra series resistance is a source of wasted energy; large sense resistors can lead to excessive heat dissipation, with potential long-term reliability concerns.

Is there any reason to use a large sense resistor? The primary advantage is to increase the over-all output voltage (EQ1). This can be useful when the amplifier has fixed gain or limited gain configurability.

$$\begin{array}{c} \text{CURRENT-SENSE} \\ \text{GAIN} \\ \text{V}_{\text{OUT}} = \text{GAIN}_{\text{AMP}} * \text{R}_{\text{SENSE}} * \text{I}_{\text{SENSE}} \end{array} \quad [\text{EQ1}]$$

There is a limit to the size of the sense resistor. The input range of the amplifier and the maximum expected current will determine the largest practical sense resistance (EQ2).

$$\text{R}_{\text{SENSE_MAX}} = (\text{V}_{\text{SENSE_MAX}} / \text{I}_{\text{SENSE_MAX}}) \quad [\text{EQ2}]$$

As an example, if 50mA is the maximum expected current through the sense resistor ($\text{I}_{\text{SENSE_MAX}}$) and the high side current sense amplifier can accept inputs up to 250mV ($\text{V}_{\text{SENSE_MAX}}$), the maximum sense resistance is 50hms ($\text{R}_{\text{SENSE_MAX}}$).

Ideally, the designer should not be forced to add sense resistance to compensate for the amplifier. As long as the amplifier can operate with sufficient gain and gain accuracy, the designer should instead focus on the minimum acceptable sense resistance. This can be calculated from the current sense amplifier's input offset voltage and the smallest current that must be resolved:

$$\text{R}_{\text{SENSE_MIN}} = (\text{V}_{\text{OFFSET}} / \text{I}_{\text{RES}}). \quad [\text{EQ3}]$$

As an example, if 1mA resolution is required (I_{RES}) and the offset voltage of the high-side current sense amplifier is 1mV (V_{OFFSET}), the minimum sense resistance is 10hm ($\text{R}_{\text{SENSE_MIN}}$). Equation 3 highlights a key point: the minimum sense resistance is directly related to the offset of the high side current sense amplifier.

TAKING A CLOSER LOOK AT A MODERN CURRENT SENSE AMPLIFIER

With high-side current sensing in mind, precision high-side current sense amplifiers offer dramatic improvements in performance over previous generations. For example, Linear Technology's LTC6102 is a new high-side current sense amplifier that incorporates zero-drift technology.

This amplifier has an input offset voltage of only 10μV and an offset drift of 50nV/°C Max. Compared to previous generations of current sense amplifiers, the LTC6102 can use a significantly smaller sense resistor. If the system can tolerate a large V_{SENSE} , the LTC6102 can accept sense voltages up to 2V. The combined offset plus this maximum sense voltage provides over 106dB of dynamic range, allowing the LTC6102 to resolve microamps from amps of current. Sensing very small current

To address the challenges of current sensing, high-side current sense amplifiers were created

is possible since any gain can be selected with external resistors. The gain accuracy can be better than 99% by using precision resistors.

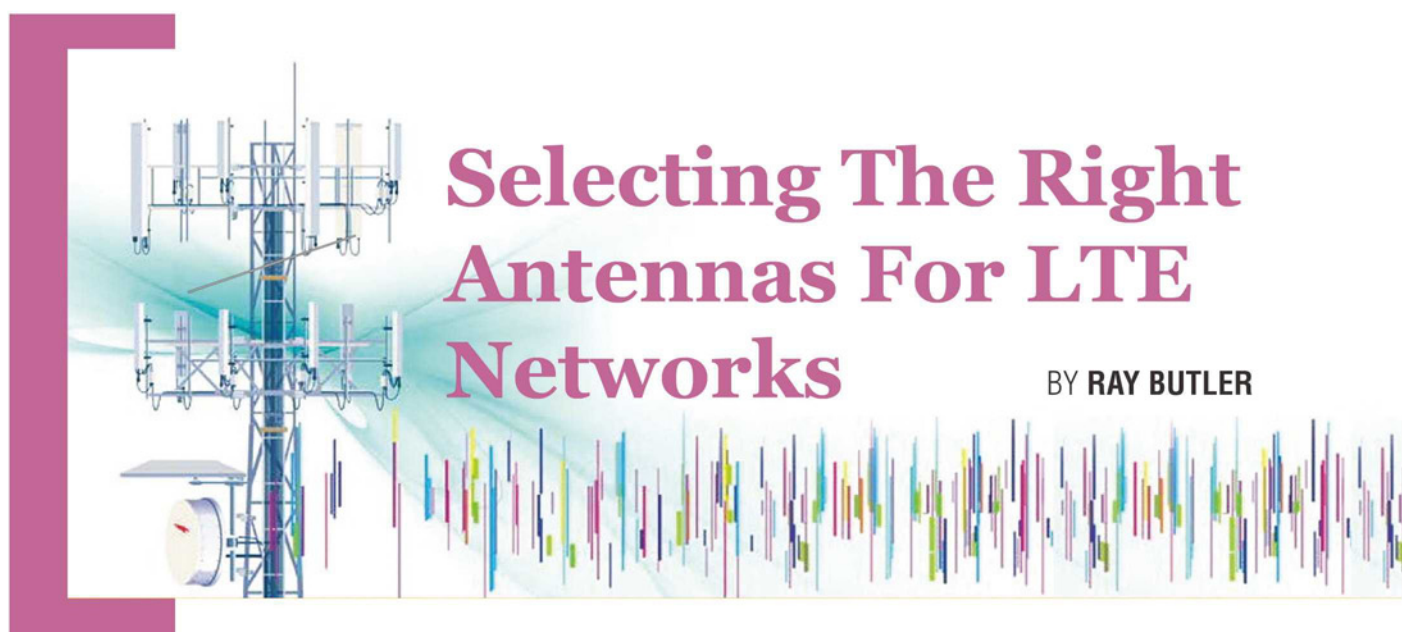
The LTC6102 does not compromise on other important current sense features, either. The high impedance inputs limit the input bias current to less than 300pA. The LTC6102 can operate with an input common mode voltage up to 105V. The common mode rejection of 130dB contributes less than 32μV of offset error across a full 100V of input common mode voltage range. For fault protection, the LTC6102 has a 1μsec response time, allowing it to quickly initiate a power shutdown in the event of unexpected load or supply changes.

CONCLUSION

Precision high-side current sense amplifiers offer inherent benefits for monitoring and controlling current. Advancing technologies in battery management and motor control, to name a few examples, are creating a significant demand for current sense amplifiers with higher common mode voltages, higher accuracy, and better precision. Leading the way, the LTC6102 breaks new ground with an impressive set of features and outstanding precision characteristics. High-side current sense amplifiers have now achieved the performance levels of industry leading precision op-amps, giving designers a simple, versatile and highly accurate alternative to the lower precision or more complicated current sense circuits of the past.

For more information about current sensing, Linear Technology has compiled the I_{SENSE} Application Note, an extensive collection of current sense circuits. Download it now at www.linear.com/currentsense

**Linear Technology (UK) Ltd • Tel: 01628 477066
Email: uksales@linear.com • www.linear.com**



Selecting The Right Antennas For LTE Networks

BY RAY BUTLER

Rapid mobile data growth requires the wireless industry to use more sophisticated, higher-capacity access technologies like LTE. While supporting many advanced antenna techniques, LTE needs precise containment of the radio frequency (RF) signals used to transmit mobile data. Network operators increasingly need high-performance antennas to meet these performance metrics.

With the many antenna techniques LTE supports – and the often highly different antenna design requirements, as well as different network applications – it can be challenging to select the best antenna to fit a given scenario.

Constructing high-performance wireless networks is like building a pyramid, wherein the lower layers of precisely-cut stones support the above layers and any inconsistency can seriously harm the entire structure.

In wireless network optimization, the bottom stones are the physical layer, which is the network equipment that delivers the RF coverage. Operators need to build and optimize the physical layer to set a solid foundation on which they can add layers for

parameter optimization, radio resource management and other advanced features.

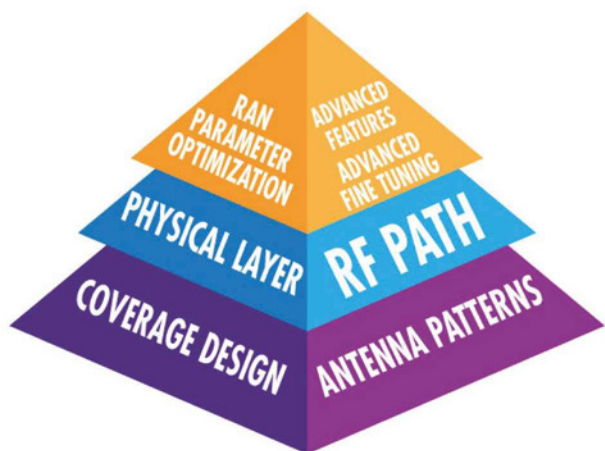
The RF coverage layer must be finely sculpted for LTE networks to run optimally. The upper layers will not reach their full potential without that well-sculpted, finely-calibrated foundation. A poorly sculpted base can lead to reduced throughput that diminishes the quality of service for subscribers. Operators must deliver strong RF signal strength in the desired coverage areas with sharp attenuation of that signal outside each area.

Deciding which basestation antenna to use is a critical decision for achieving finely-sculpted RF patterns. Antenna technology advances including MIMO (multiple input multiple output), beamforming and the use of various transmission modes plus RF path architecture changes complicate that decision. The addition of more frequency bands for LTE has forced an increase in the number of antenna ports, and antenna sizes have grown as more capabilities are integrated. All these changes impact antenna performance and selection.

For example, MIMO technology, or spatial multiplexing, increases throughput by transmitting distinct data streams over different signal paths using the same resources in both frequency and time. MIMO requires a high signal-to-interference-plus-noise ratio (SINR) and low correlation of each path. The signal de-correlation is a result of the antennas (polarization or spatial diversity) and the environment (rich scattering).

There are several types of MIMO: single-user MIMO (SU-MIMO), multi-user MIMO (MU-MIMO), open loop and closed loop MIMO and massive MIMO. The various types of MIMO are a function of which LTE transmission mode is selected.

SU-MIMO requires multiple antennas at both ends of the link to spatially multiplex channels serving a single user. SU-MIMO is most often used on the downlink as there are antenna and power limits to device designs.



MU-MIMO combines multiple spatially “de-correlated” users onto the same resources. MU-MIMO does not increase peak user throughput, but it does increase average user throughput and sector capacity. Both uplink and downlink MU-MIMO are possible. Massive MIMO uses a two-dimensional array of closely-spaced antennas, MU-MIMO with tens of devices, and is expected to possibly use an active antenna system (AAS).

Beamformers use an array of antenna elements individually phased in such a way as to form beams (or nulls) in a desired direction. Typical beamforming antennas have highly-correlated, closely-spaced elements and columns. Passive antennas can support horizontal beamforming. An AAS integrates the active transceiver array and the passive antenna array into the same

“The addition of more frequency bands for LTE has forced an increase in the number of antenna ports, and antenna sizes have grown as more capabilities are integrated

radome, and supports two-dimensional (azimuth) and 3D (both azimuth and tilt) antenna array configurations.

Operators must choose antennas based on the techniques most advantageous to their network design and needs. Operators with frequencies below 1GHz are most likely to use a single-column cross-polarized antenna due to size limitations inherent with leasing and zoning (typically less than or equal to 610mm wide x 1830mm high), while antennas serving the 1-2GHz range are typically limited today to no more than two cross-polarized columns. Size constraints likewise limit 2.5GHz four-column (eight-port) antennas to a column spacing $\leq 0.65\lambda$ (0.65 wavelengths), which can give sub-optimal uplink performance compared to 1λ , but serves the downlink well. ●

Ray Butler is Vice President of Wireless Network Engineering at CommScope.

This column is an edited extract from CommScope's new eBook, 'LTE Best Practices'. Over the next few months different authors of this eBook will contribute articles to this section.

SPECIAL OFFERS
for full sales list
check our website

www.stewart-of-reading.co.uk
Check out our website, 1,000's of items in stock

Used Equipment – **GUARANTEED**
All items supplied as tested in our Lab
Prices plus Carriage and VAT

IFR 2025	Signal Generator 9kHz - 2.51GHz Opt 04/11	£1,250	Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/S	£350
Fluke/Philips PM3092	Oscilloscope 2+2 Channel 200MHz Delay etc	£295	Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
HP34401A	Digital Multimeter 6.5 digit	£325	R&S APN62	Syn Function Generator 1Hz-260KHz	£225
Agilent E4407B	Spectrum Analyser 100Hz - 26.5GHz	£5,000	R&S DPSP	RF Step Attenuator 139dB	£300
HP3325A	Synthesised Function Generator	£195	R&S SMR40	Signal Generator 10MHz - 40GHz with Options	£13,000
HP3561A	Dynamic Signal Analyser	£650	Cirrus CL254	Sound Level Meter with Calibrator	£40
HP3581A	Wave Analyser 15Hz - 50KHz	£250	Farnell AP60/50	PSU 0-60V 0-50A 1KW Switch Mode	£195
HP3585B	Spectrum Analyser 20Hz - 40MHz	£1,500	Farnell H60/50	PSU 0-60V 0-50A	£500
HP53131A	Universal Counter 3GHz	£600	Farnell B30/10	PSU 30V 10A Variable No Meters	£45
HP5361B	Pulse/Microwave Counter 26.5GHz	£1,250	Farnell B30/20	PSU 30V 20A Variable No Meters	£75
HP54600B	Oscilloscope 100MHz 20MS/S	from £125	Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
HP54615B	Oscilloscope 2 Channel 500MHz 1GS/S	£650	Farnell LF1	Sine/sq Oscillator 10Hz-1MHz	£45
HP6032A	PSU 0-60V 0-50A 1000W	£750	Racal 1991	Counter/Timer 160MHz 9 Digit	£150
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350	Racal 2101	Counter 20GHz LED	£295
HP6624A	PSU 4 Outputs	£350	Racal 9300	True RMS Millivoltmeter 5Hz-20MHz etc	£45
HP6632B	PSU 0-20V 0-5A	£195	Racal 9300B	As 9300	£75
HP6644A	PSU 0-60V 3.5A	£400	Black Star Orion	Colour Bar Generator RGB & Video	£30
HP6654A	PSU 0-60V 0-9A	£500	Black Star 1325	Counter Timer 1.3GHz	£85
HP8341A	Synthesised Sweep Generator 10MHz-20GHz	£2,000	Ferrograph RTS2	Test Set	£50
HP83731A	Synthesised Signal Generator 1-20GHz	£2,500	Fluke 97	Scopemeter 2 Channel 50MHz 25MS/S	£75
HP8484A	Power Sensor 0.01-18GHz 3nW-10uW	£125	Fluke 99B	Scopemeter 2 Channel 100MHz 5GS/S	£125
HP8560A	Spectrum Analyser Synthesised 50Hz - 2.9GHz	£1,950	Fluke PM5420	TV Gen Multi Outputs	£600
HP8560E	Spectrum Analyser Synthesised 30Hz - 2.9GHz	£2,400	Gould J3B	Sine/sq Oscillator 10Hz-100KHz Low Distortion	£60
HP8563A	Spectrum Analyser Synthesised 9KHz-22GHz	£2,750	Gould OS250B	Oscillator Dual Trace 15MHz	£50
HP8566B	Spectrum Analyser 100Hz-22GHz	£1,600	Gigatronics 7100	Synthesised Signal Generator 10MHz-20GHz	£1,950
HP8662A	RF Generator 10KHz - 1280MHz	£1,000	Panasonic VP7705A	Wow & Flutter Meter	£60
HP8970B	Noise Figure Meter	£750	Panasonic VP8401B	TV Signal Generator Multi Outputs	£75
HP33120A	Function Generator 100 microHz-15MHz - no moulding handle	£295	Pendulum CNT90	Timer Counter Analyser 20GHz	£995
Marconi 2022E	Synthesised AM/FM Signal Generator 10KHz-1.01GHz	£325	Seaward Nova	PAT Tester	£125
Marconi 2024	Synthesised Signal Generator 9KHz-2.4GHz	£800	Solartron 7150	6 1/2 Digit DMM True RMS IEEE	£65
Marconi 2030	Synthesised Signal Generator 10KHz-1.35GHz	£750	Solartron 7150 Plus	as 7150 plus Temp Measurement	£75
Marconi 2305	Modulation Meter	£250	Solartron 7075	DMM 7 1/2 Digit	£60
Marconi 2440	Counter 20GHz	£295	Solartron 1253	Gain Phase Analyser 1mHz-20KHz	£750
Marconi 2945	Communications Test Set Various Options	£2,500	Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Marconi 2955	Radio Communications Test Set	£595	Thurby PL320	PSU 0-30V 0-2A Digital	£50
Marconi 2955A	Radio Communications Test Set	£725	Thurby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65
Marconi 2955B	Radio Communications Test Set	£850	Wavetek 296	Synthesised Function Generator 2 Channel 50MHz	£450
Marconi 6200	Microwave Test Set	£1,950			
Marconi 6200A	Microwave Test Set 10MHz-20GHz	£2,500			
Marconi 6200B	Microwave Test Set	£3,000			
IFR 6204B	Microwave Test Set 40GHz	£10,000			
Marconi 6210	Reflection Analyser for 6200 Test Sets	£1,250			
Marconi 6960B with	6910 Power Meter	£295			
Marconi TF2167	RF Amplifier 50KHz - 80MHz 10W	£75			
Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/S	£800			

STEWART OF READING

17A King Street, Mortimer, Near Reading, RG7 3RS
Telephone: 0118 933 1111 • Fax: 0118 933 2375
9am – 5pm, Monday – Friday
Please check availability before ordering or **CALLING IN**



Graphic Object Layer

LUCIO DI JASIO, ELECTRONICS ENGINEER AND TECHNICAL AUTHOR, PRESENTS THIS SERIES ON EMBEDDED USER-INTERFACE DESIGN ON A BUDGET

In the previous seven instalments of this column, we analysed all the parts of the Mikromedia board. For each of them we found out which module of the Microchip Library for Applications (MLA) was available and how it could be configured to match the Mikromedia specific interfaces. By now you would have realised this mostly involves adding a few lines to the HardwareProfile.h file and eventually commenting or uncommenting a few lines in selected config files to enable/disable optional features. But in the development of user interfaces (UI) for true embedded applications, these building blocks can only take us so far. To develop large and complex interfaces with a professional look and feel we need to learn to use a more powerful (second) layer of abstraction – enter the graphic object layer (GOL).

Of Objects And Widgets

Sitting on top of the graphics primitives, the touch interface and graphics resources (fonts and images), we find the MLA GOL module, where we finally meet for the first time the “widgets” – the graphical components we have all grown to recognize as the essence of all graphical user interfaces (GUIs).

The name of this library, however, can be a bit misleading: does it

mean we need to use an object-oriented programming language; is C++ now a requirement; and so on?

I am happy to report it is not – at least not with the MLA GOL library! An object-oriented design methodology is not a function of the language we use. The GOL library is, in fact, written entirely in C language and compatible with the MPLAB XC compiler suite.

This means that we can design a GUI for a 16-bit microcontroller (PIC24, dsPIC33) and quickly recompile it for a 32-bit model (to be used on a PIC32 Mikromedia) or even for an 8-bit microcontroller (on a PIC18 Mikromedia).

In perfect object-oriented style, the GOL library requires us to create objects (widgets) by defining their type, position and a number of additional attributes using a simple BtnCreate() function (see Listing 1).

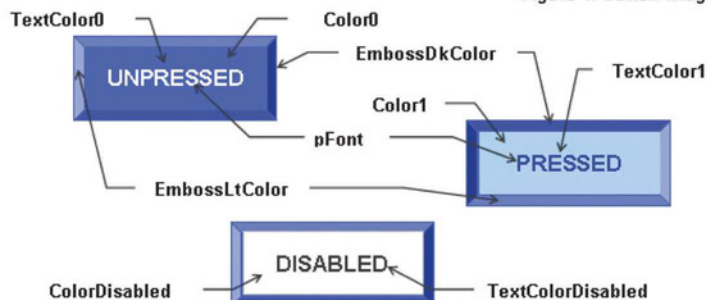
```
pButton = BtnCreate(
10,                // object's ID
10, 10, 110, 60,  // left, top, right, bottom coordinates
0,                // radius of rounded edge, use pointy
                  edges
BTN_DRAW,         // draw the object after creation
NULL,             // no bitmap used
"Done",           // text to be shown inside the button
NULL              // use default style scheme
);
```

Listing 1: Example of a *button* widget

There are similar xxxCreate() functions for all possible types of widgets (there were 21 at last count), all taking a similar list of arguments.

Then we talk to the widgets, and they communicate with each other by exchanging messages, which are simple, standardized, data structures that carry information about an event and optional parameters (see Listing 2).

Figure 1: Button widget



CommonBkColor – used to hide the button on the screen.

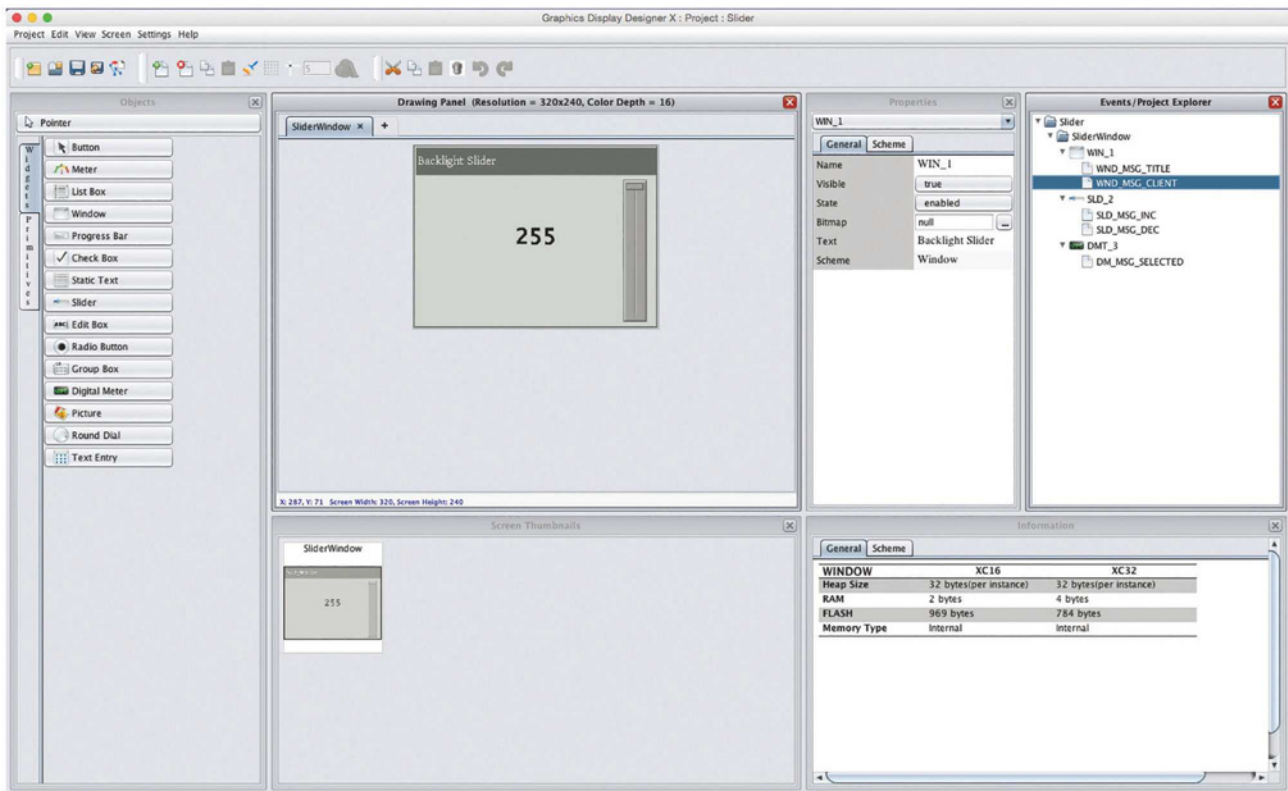


Figure 2: Screenshot of the GDDX user interface

```
typedef struct {
    BYTE type;
    BYTE uiEvent;
    SHORT param1;
    SHORT param2;
} GOL_MSG;
```

Listing 2: A message data structure

User touch inputs are a great example of such messages. The connection between the touchscreen module and the GOL, for example, is achieved by a single function, `TouchGetMessage()`, that checks for the last touch event (coordinates) and packs them inside a message data structure (see Listing 3 below).

```
void TouchGetMsg(GOL_MSG *pMsg)
{
    static SHORT prevX = -1;
    static SHORT prevY = -1;
    SHORT x, y;
    x = TouchGetX();
    y = TouchGetY();
    pMsg->type = TYPE_TOUCHSCREEN;
    ...
    if ((x != -1) && (y != -1))
    { // Pressed
        pMsg->uiEvent = EVENT_PRESS;
    }

    ...
}
```

Listing 3: Snippet of the `TouchGetMsg()` function

Page Layout

If we organize the GUI in pages, similar to mobile applications, there is simply too little room to subdivide the screen further into smaller windows, so each page is completely defined by an active widgets list. The GOL main loop is constantly scrolling down the active widgets list and passing around messages (response). Each widget can decide to act upon selected messages and/or send messages back. Each widget knows its position on the screen and is ready to paint itself (!) when asked to do so by receiving a paint message.

There are entire books devoted to teaching the art of optimal layout of the graphical elements on a page, so we definitely won't delve into the subject here, except to point out how tedious the work can quickly become, because for each widget on each page we have to specify the absolute position; see the second row of parameters: left, top, bottom, right, passed to the `BtnCreate()` function in Listing 1.

GDD X Comes To The Rescue

As we have seen previously (you will remember the graphic resource converter, or GRC), the MLA developers have created a handy tool, the Graphic Display Designer (GDD), to help us prototype and rapidly develop GOL user interfaces without spending hours "counting pixels"; see Figure 2.

As was the case of the GRC, the tool is written in Java and therefore works across all platforms (Windows, Linux and OS X) but also, in the most recent incarnation it can be installed directly as a plugin inside MPLAB X (hence the new name GDD X).

Not only the manual layout of widgets and the definition of multiple pages is simplified, but the tool attempts also to automate the definition of the events (and the messages produced) to connect widgets according to the application logic. This requires a little more explaining.

I Will Call You Back

To keep the overall application design simple and modular, rather than defining actions as separate functions for each widget, the GOL library groups all activity in two “loops”, the message passing loop and the drawing loop. The message passing loop does exactly what you’d expect, taking each event/message generated by the application and passing it to each widget in the active list (page). A number of automatic actions/reactions are predetermined (built in) for each widget type. If you send a touch press event with a pair of coordinates that correspond exactly to the location of a button on the screen, for example, it will trigger that button to change state. The drawing loop, then, is where each widget is asked to redraw itself (if necessary) to produce the required visual feedback. In the example of the button being pressed, this will force the button widget to redraw itself to look “pressed”.

On top of such automatic behaviours, though, it is up to the user to inject additional application specific logic. For example, we might want to associate moving a slider on the screen to a corresponding change of intensity of the display backlight. To allow this, a special user defined function (callback) is called inside each loop: `GOLMsgCallback()` and `GOLDrawCallback()` respectively.

```
WORD GOLMsgCallback( WORD objMsg, OBJ_HEADER*
pObj, GOL_MSG* pMsg)
{
if( pObj->ID == 1) // intercept messages from the slider
{
// update the screen backlight
BacklightSet( SldGetPos( pObj));
}
}
```

Listing 4: `GOLMsgCallback()` example

The example in Listing 4 shows how to intercept a message coming from a slider widget (recognized by its specific ID assigned at creation) and then change the display backlighting based on the position of the slider.

Similarly, by intercepting the drawing loop using the `GOLDrawCallback()` function we can define custom graphic effects that go beyond the normal expected behaviour of the stock widgets.

The Heap

While playing with dynamic memory allocation in an embedded application can be a bad idea, it must be noted that the GOL library makes very judicious use of memory and, in fact, there is very little dynamic involved too. It is just a matter of flexibility and convenience. The library does not require allocating global buffers of pre-computed size for the widgets we will use, so it does it for us using the standard

Sitting on top of the graphics primitives, the touch interface and graphics resources (fonts and images), we find the MLA GOL module

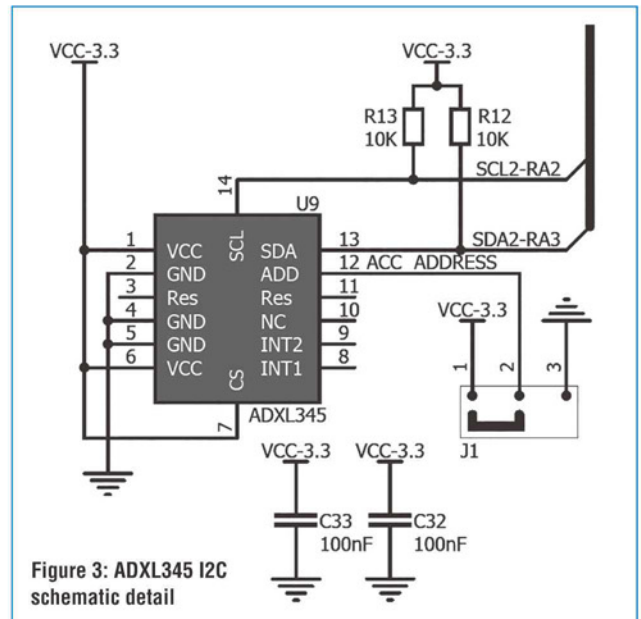


Figure 3: ADXL345 I2C schematic detail

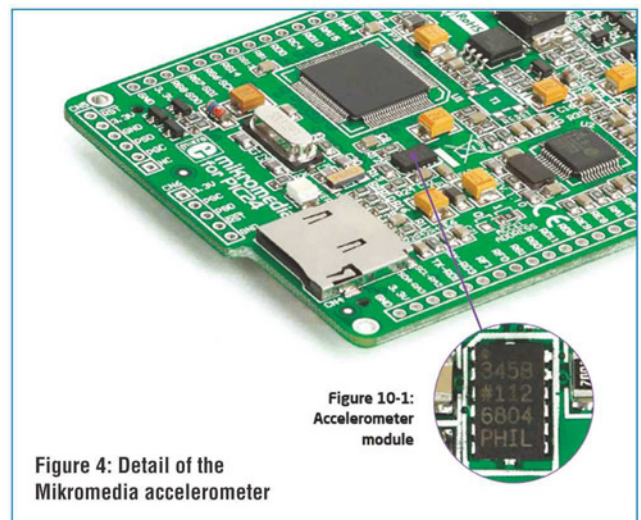


Figure 4: Detail of the Mikromedia accelerometer

C heap management library. It is then entirely up to us to “manage” the memory, attempting to free and re-use chunks of it as we move from page to page in our application, or we rather prefer to keep things in place (so to speak) to avoid the hassle and the complications that derive from it.

In any case, we have to reserve a sufficient amount of (RAM) memory for Heap use by entering it in the MPLAB XC linker (XC16-ld) project options. GDDX does compute automatically the sum of all the memory required for the application, making this step simple and guess-free.

Accelerate The Applications

There is one last peripheral module of the Mikromedia board that we have not yet discussed: the accelerometer. This is an ADXL345 device connected via I2C interface in each of the Mikromedia boards;

it can be considered just another user input sensor to integrate in the application interface.

Figure 5 shows a demo application representing the Z axis sensor output using a 'round dial' widget, and the X and Y axes using two 'progress bar' widgets.

Now the true fun begins, as you dig deeper into these libraries and start applying this knowledge to other boards and designs of your own creation. ●

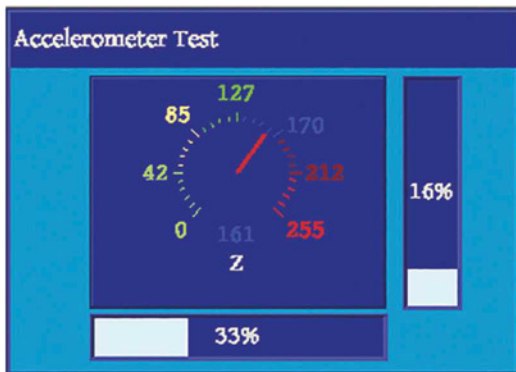


Figure 5: The ADXL345 accelerometer user interface

USER INTERFACE DESIGN FOR EMBEDDED APPLICATIONS

Lucio Di Jasio is EMEA Business Development Manager at Microchip Technology. He has held various technical and marketing jobs in the company's 8, 16 and 32-bit divisions for the past 18 years.

Lucio has published several books on

programming for embedded control applications, and we have three copies of his book 'Graphics, Touch, Sound and USB, User Interface Design for Embedded Applications' to give away at the end of this series.

If you want to win this book, please send an email to svetlanaj@sjpbusinessmedia.com, mentioning the title in the heading.



UK designed,
UK made,
with pride.

Tel. 01298 70012
www.peakelec.co.uk
sales@peakelec.co.uk

Atlas House, 2 Kiln Lane
Buxton, Derbyshire
SK17 9JL, UK

Follow us on twitter
for tips, tricks and
news.
[@peakatlas](https://twitter.com/peakatlas)

For insured UK delivery:
Please add £3.00 inc VAT
to the whole order.
Check online or
give us a call for
overseas pricing.

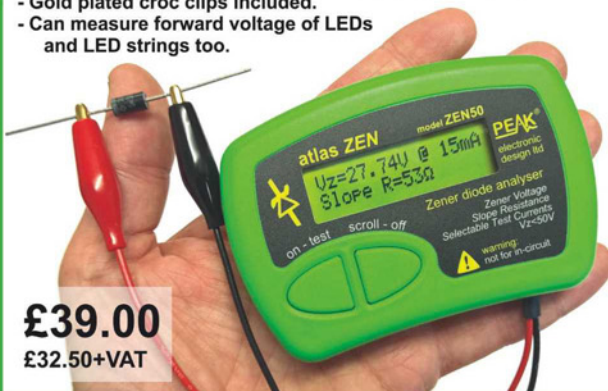
PEAK[®]
electronic design ltd

ZEN50 Zener Diode Analyser (inc. LEDs, TVSs etc)

Brand new product!

Introducing the new **Atlas ZEN** (model ZEN50) for testing Zeners (including Avalanche diodes) and many other components.

- Measure Zener Voltage (from 0.00 up to 50.00V!)
- Measure Slope Resistance.
- Selectable test current: 2mA, 5mA, 10mA and 15mA.
- Very low duty cycle to minimise temperature rise.
- Continuous measurements.
- Single AAA battery (included) with very long battery life.
- Gold plated croc clips included.
- Can measure forward voltage of LEDs and LED strings too.



£39.00
£32.50+VAT

PCA23 SOT23 Test Adapter for Multimeters and Peak Analysers

Brand new product!

Designed to look like a giant SOT23 device, this beautiful adapter allows you to easily connect surface mount diodes, Zeners, MOSFETs and transistors.

Just place your device into the controlled-force clamshell and connect your probes to the gold plated test points.

Great for connecting crocs, hook-probes and multimeter prods.



£19.80
£16.50+VAT

DCA75 DCA Pro

The famous "A very capable analyser"

- Detailed review in *RadCom* magazine

Exciting new generation of semiconductor identifier and analyser. The **DCA Pro** features a new graphics display showing you detailed component schematics. Built-in USB offers amazing PC based features too such as curve tracing and detailed analysis in Excel. PC software supplied on a USB Flash Drive. Includes Alkaline AAA battery and comprehensive user guide.



NEW LOW PRICE!
£105.00
£87.50+VAT

It's only possible to show summary specifications here. Please ask if you'd like detailed data. Further information is also available on our website. Product price refunded if you're not happy.



Idle State

BY MYK DORMER

"G

reen design means ultra-low power". Could I have crammed any more popular buzzwords into one short sentence?

For good or ill, the ultimate power consumption of our products is fast becoming one of the most important factors in their overall design, for reasons as abstract as "ecological goodness" or

as practical as the inconvenience and/or expense of battery replenishment in portable devices.

Unfortunately, along with the desire for lower power comes the desire for more functionality. More processing power, more functions and, in the low-power radio arena, higher transmit power, better receiver performance and greater data throughput. More sophisticated technology helps to some degree. Smaller chip geometries promise improved "megahertz per milliamp" figures, and new display types reduce wasted power there – but receiver large-signal performance needs current in the front-end stages, and a transmitted watt of power is still a watt, and no power amplifier will exceed 100% efficiency.

In low-power radio design the upshot of this is that much, or even most, of the critical circuitry in a battery-powered or otherwise power-critical design is duty-cycled on and off to reduce the average current (once conventional design has minimised the peak value), a technique pioneered by the paging industry. Frequently, this duty cycle can be a fraction of 1% or less, and can reduce a peak radio current of several hundred milliamps to no more than a few tens of microamps, average.

This is a very well understood technique, used in a wide variety applications – including the mobile phone, but it still harbours an interesting trap for the unwary: In designing a low duty cycle system it is necessary to control the cycle rate by means of some piece of hardware that is always "on". This is usually a low-frequency oscillator of some type, driving either a hardware digital counter or the "interrupt out of sleep" function of the device's main processor. Common alternatives include RC oscillators internal to the processor, or in timing-critical applications, a low frequency 'watch' crystal, combined with a CMOS oscillator.

Unless the design is very simplistic and powers down all the circuitry in the off state, then there will be a situation where

the controlling hardware is powered (even if most of it is in some form of sleep mode) and significant parts of the rest of the circuit is either powered down or in standby.

Now, these standby modes ("chip select false") are very useful, but often either do not reduce the current drain sufficiently or actually (internally) the standby is a 'power off', which switches off significant parts of the interface circuitry.

That leaves us with the most absolute case: the high-current circuitry, such as a radio module, an LCD display or a complex signal processing chip, that has its power supply rail switched off (disconnected) when "off". It would seem that, when the actual power rail is absent (by means of an inline switch device, or the shutdown of a dedicated regulator), there could be no possible difficulty – which is where the trap lies!

Consider the likely circuit configuration. The powered controlling device (probably not the main processor) not only controls the rail switching but also has one or more control or signal lines feeding across the powered/unpowered barrier. If conventional design methodologies have been followed, there is

" The design must be deliberately arranged so that the powered section of the unit never presents a voltage to an input or an output of the unpowered part, although this is easier said than done

a very good chance that some of those lines will be "active low"; for example: chip select pins are often active low and "UART output" serial ports likewise idle in their mark (high) state.

This can be a disaster. Active low signals idle at a (logic) high voltage. The outputs of the unpowered circuitry will no longer be able to present this idle state to the inputs of the controlling circuit, causing the potential for spurious activations, serial UART framing/over-run errors and so on.

If there are pull-ups on the powered side, or if the lines are outputs from that section, then things can be worse: the voltage present on the (innocent, inactive) output will

be present on the input pin of the unpowered circuit. In the best case this can just result in wasted current flowing but more likely will cause the substrate diodes in the unpowered circuit's logic inputs to conduct, allowing unexpected parts of the circuit to power up, latch up, or possibly fail to initialise properly when "properly" powered up in the operating cycle. In some cases, I have seen entire integrated circuits powered up through leakage current flowing into just one input. At this point wasted current is the least of your worries.

The cure is as obvious as it is draconian. The design must be deliberately arranged so the powered section of the unit never presents a voltage to an input or (by direct logic drive or via a pull-up) an output of the unpowered part, although this is easier said than done.

Where the option exists to define the state of the control or data lines, such as, for example, where a data port is being handled in software between two processors, then it is necessary to ensure that all the idle states are low – data ports set to zero before power-off, and all control signals and strobes defined as active high.

If the unpowered circuit's input is unavoidably active low,

then add an open collector (or drain) buffer, or ensure that the driving output in the powered side of the circuit is switched to a high-impedance ("tri-state") condition before powering off. In either case a pull-up is necessary to the switched power rail on the unpowered side.

Lastly, if the unpowered circuit presents an output to the powered circuit that usually idles in a high state (like a UART output), then a high-value pull-down is needed to avoid an undefined logic state on the input, and the powered circuit input will either need to be programmed to ignore the apparent spurious state during power-off, or it can be isolated with a CMOS switch or a logic (OR) gate.

This is, I admit, an elementary mistake to run into, but sometimes it's those very simple issues that complicate much more complex jobs, if they are not spotted in time.


Even after thirty years on the bench, I have still caught myself missing this a time or two in the recent past.

Hopefully, now you won't. ●

Myk Dormer is a Senior RF Design Engineer at Radiometrix Ltd www.radiometrix.com

UMT-H

SMD fuse with high breaking capacity



- High breaking capacity of 1500 A
- High rated voltages up to 277 VAC / 250 VDC
- 19 rated currents from 160 mA to 10 A
- Compact, square design: 5.3 x 16 mm

umt-h.schurter.com

SCHURTER
ELECTRONIC COMPONENTS



HARMONIC

SOFTWARE SYSTEMS

WOULD YOU BENEFIT FROM :

- On site expertise to get your projects back on track?
- Agile Engineering Services for VxWorks and Embedded Linux?
- Years of experience with multiprocessing, hypervisors, hardware virtualization, networking, device drivers and board support packages?
- Embedded Security for your next IoT device?



CUSTOMER TESTIMONIAL:

"Their knowledge of the Wind River tools and VxWorks RTOS was phenomenal. This combined with their structured approach to fault isolation and processor knowledge made their efforts invaluable to us."



VISIT US AT STAND G35,
EMBEDDED DESIGN SHOW,
21ST-22ND OCTOBER,
RICOH ARENA, COVENTRY

WWW.HARMONICSS.CO.UK
SALES@HARMONICSS.CO.UK
01403 784500

**VISIT OUR
WEBSITE FOR A
QUICK QUOTE**

EMBEDDED AND APPLICATION DEVELOPERS – LEARNING FROM EACH OTHER

BY **GREG LAW**, FOUNDER AND CEO, UNDO SOFTWARE



vidence of software's growing importance is everywhere: from industrial devices to those in the home, products and processes increasingly rely on software for controlling operations and introducing new features to differentiate them from the competition. This trend is having a major impact on

software development, with embedded systems now combining traditional device-level software with application code. The graphical user interface (GUI) on a smart TV is a perfect example – previously it would have been more basic, and now its look and usability are key selling points in a competitive market.

The Evolution Of Consumer Electronics

Another good example of how things have changed is the way we listen to music. The original Sony Walkman of 1979, which played cassette tapes, was purely mechanical, with no software involved. As it evolved, an LCD was added to it, requiring an 8- or 16-bit embedded microcontroller, with code running on the bare metal of the device.

Next, as functionality and complexity of music devices increased, an embedded real-time operating system (RTOS) was required to manage drivers, schedulers and sensors. Most software was still system-level rather than application code, but the balance was beginning to shift.

Fast-forward to the iPod Touch and the device is running a full-blown operating system that manages memory, scheduling, user interfaces and application code. When we get to the iPhone, the roles of embedded and application code have swapped around completely from the early iPods. The OS takes up the minority of code on the device; most is application code, which also drives the touchscreen-based user experience.

Essentially, in 20 years the picture has changed around completely – and not just in consumer electronics. Previously hardware focused devices, from uninterruptible power supplies to industrial robots, now rely on application software to provide user-visible differentiation, a trend that should continue. The rise of the Internet of Things (IoT) and the spread of software in sectors such as connected cars means that software will run all aspects of our world, and the vast majority of that software will be application-level code.



The rise of the Internet of Things and the spread of software in sectors such as connected cars means that software will run all aspects of our world, and the vast majority of that software will be application-level code

The Changing Debugging Challenge

The ways embedded and application software is written and debugged traditionally have been very different. Tracking down a bug depends on three things – visibility of code, the time between the bug striking and a problem being detected, and how deterministic the failure is, e.g. does the failure unfold in exactly the same way from the same starting state or does it vary due to outside factors.

These scenarios are, however, not identical for embedded and application developers. Embedded developers working in tightly constrained environments have had to contend with a lack of visibility into their code and what it's doing. There was normally little clue as to what led to something going wrong or where an issue originated, making it correspondingly hard to track down when creating hardware. For example, when writing code on bare metal or to bring up a device, we don't even have a printf to rely on. Consequently, the Joint Test Action Group (JTAG)

created an IEEE standard to assist with device, board and system testing, diagnosis and fault isolation. JTAG is now used as the primary means of accessing sub-blocks of integrated circuits, making it an essential mechanism for debugging embedded systems that may not have any other debug-capable communications channel. On most systems, JTAG-based debugging is available from the very first instruction after CPU reset, letting it assist with development of early boot software that runs before anything is set up.

Along with integrated circuits, entire software debugging, instruction tracing and data tracing infrastructures based around JTAG have been created for software debug around silicon architectures, from PowerPC to ARM. Processors can normally be halted, single-stepped or let run freely. Developers can set code breakpoints, both for code in RAM (often using a special machine instruction) and in ROM/Flash. Most designs have "halt mode debugging", but some allow debuggers to access registers and data buses without halting the core being debugged.

Often used in conjunction with hardware emulators, JTAG provides a rich variety of powerful debugging tools for embedded developers, including the ability to reversibly debug code; for example, debuggers can often step or run code in reverse, which is incredibly powerful when trying to track down the most elusive bugs.

Figure 2: Logical view of the full filter system, F1T pipelines

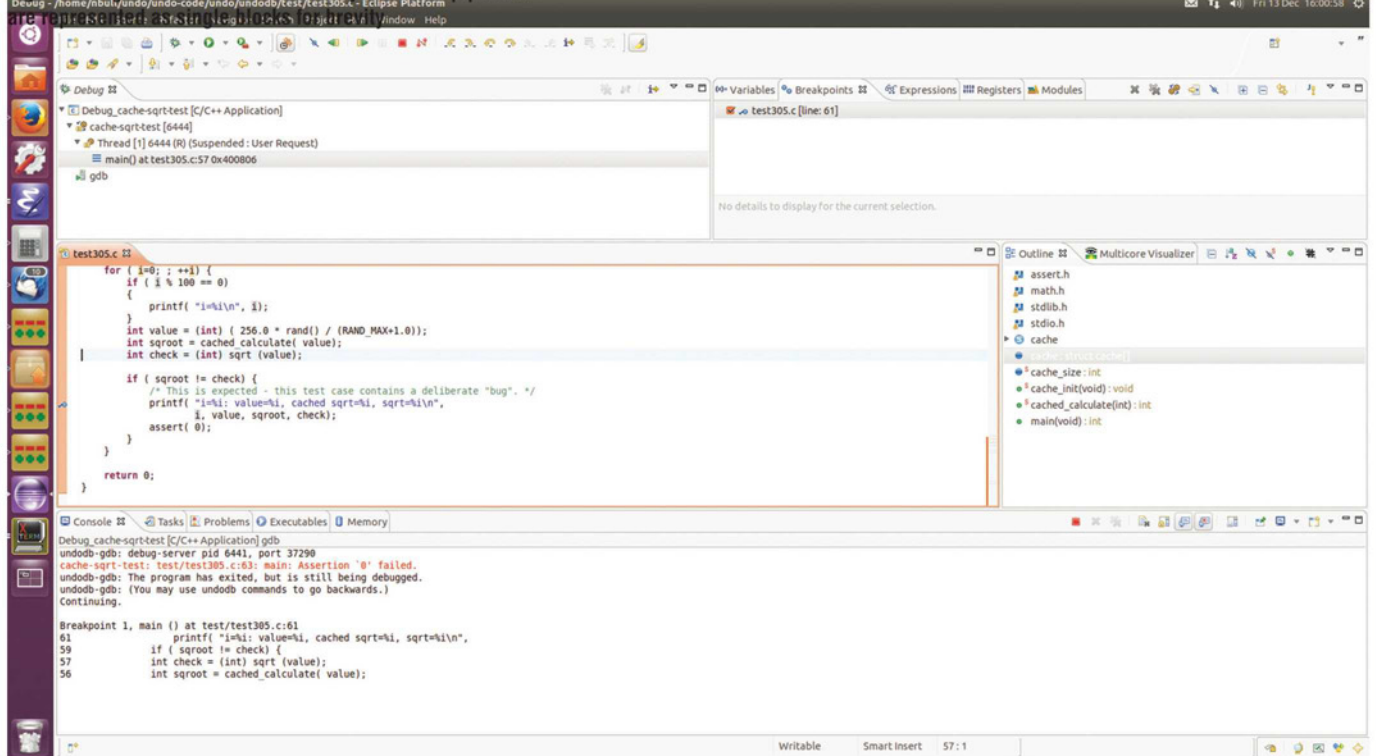


Figure 1: Reversible debugging on Eclipse through UndoDB

Debugging Application Code

In contrast, application developers have greater visibility into what their programs are doing – at least the trusty `printf` is available, plus the developer can (usually) be confident that the hardware and OS on which their code is running is stable.

However, this is outweighed by modern devices' software stacks being so much more complicated, with many more factors impacting how a program behaves. Take an Android phone app for example. It may work perfectly well on one version of the operating system on a particular phone, but exhibit bugs on others due to the fact that developers simply cannot have visibility over all potential characteristics. It is much less deterministic, with inputs from outside the program (user interaction, network sockets or system calls, for example), all potentially affecting reliability.

Essentially, it is a different sort of visibility problem, one that is getting worse as code becomes more complex and runs on more diverse devices. So it is no wonder that a study from the Judge Business School of the University of Cambridge, UK, published in 2013, found that the global cost of debugging software had risen to \$312bn annually. The study reported that developers spend 50% of their development time fixing bugs or making code work, rather than designing or writing new code. The vast majority of debugging time is spent locating the bug – once it has been found, correcting it is normally relatively simple.

The Options For Application Developers

There are a range of approaches available to help application developers with the visibility issue. These include programmatic techniques, special-case diagnosis/analysis tools, general-purpose debuggers and reversible debuggers, which enable developers to record all program activities (every memory access, every computation and every call to the operating system) and then rewind and replay to inspect the program state. Such a colossal amount of data is presented via a powerful metaphor: the ability to travel backward in time (and forward again) and inspect the program state. This enables developers to solve the mystery of what caused the bug by letting them see exactly what happened and how the code got there.

Using a reversible debugger gives visibility and more control over the code, but what to do when it is released to customers or internal QA teams? Developers need enough information about the failure to find out what went wrong. Normally this requires creating a test case and/or going back and forth with the customer to gain enough information to reproduce the error in-house. As a worst-case scenario, the developer will have to travel to the customer's site to try and track down the bug in-situ, which takes time, is costly and impacts the customer relationship.

What's needed is a way of recording what the program is

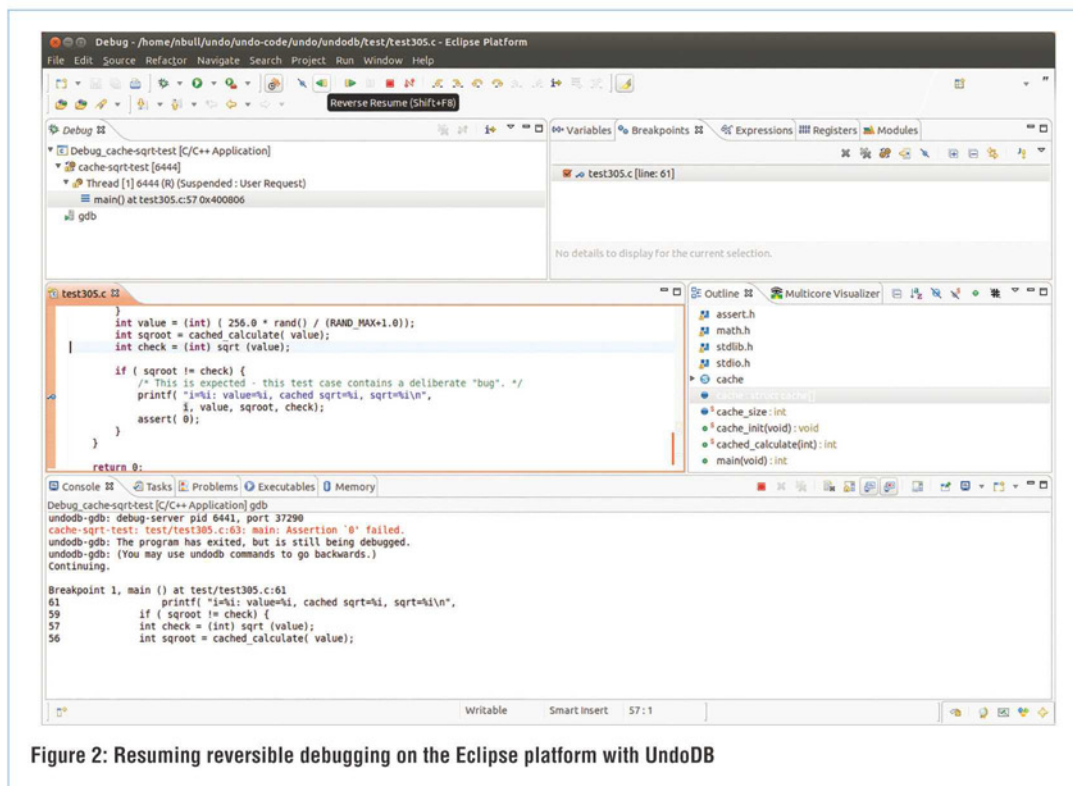


Figure 2: Resuming reversible debugging on the Eclipse platform with UndoDB

doing when it crashes, and remotely sending that back to the software vendor or internal developer, so they can see exactly what was happening, and investigate. Essentially, developers need to move from trying to reproduce the bug in-house to simply replaying it on their own machine, where it is easier and faster to investigate and solve. Tools that do this are now available to application developers.

However, due to the less constrained environments they operate in, there are far fewer debugging tools available to application developers compared with their embedded colleagues. This could partially be due to a view that many application bugs are relatively trivial and aren't worth the effort to track down, particularly if they rarely strike and cause minimal impact on the user. For example, if an Android game crashes occasionally, it is obviously less serious than a bug in the smartphone's core software that prevents the device owner making calls or picking up emails. Yet, in today's software-controlled world, an increasing number of application bugs cause critical problems, whether it is shutting down cash machines, preventing connected cars from starting, or perhaps stopping the completion of in-app purchases, which is business-critical to any company relying on them for revenue.

The Move To Unispace Development

So, how can these application bugs be tracked down more quickly and easily? The answer lies in learning from the visibility constraints within the embedded world. As mentioned earlier, most electronic devices now have a combination of embedded and application software running on them, often in tandem. While developers still sit firmly in the embedded or

application camps, they could well be within the same company, working together on the same project, with the same deadlines and resource constraints. This is an opportunity for increased sharing of best practice between the two groups.

Application developers can benefit from adopting the wide range of tools originally created for debugging embedded systems, using their power to solve their own development issues. This can give them the visibility they need to debug increasingly complex systems, even when the code is in the field. At the

very least they should take on board some of the embedded ethos that bugs are inherently bad and need to be found and eradicated, however trivial they may appear. By using embedded debugging tools, and integrating application level tools such as reversible debuggers and recording software into their workflows, they can increase visibility of code, in development or production, enabling bugs to be found and fixed faster.

For companies focusing on application development, it is time to look at how they find and fix remote bugs. Adopting tools that record what a program is doing when it crashes, and sending that information back to the software vendor or internal developer means they will be able to see exactly what is going on. Techniques such as reversible debugging to rewind and replay their code may be used to investigate issues the same way their embedded colleagues use JTAG tools on their code. This makes it simpler to quickly find and fix bugs, deliver to ever-shortening deadlines, boost overall productivity and help customer relationships.

Shifting Trends

We're moving into a world where the traditional barriers between embedded and application code are shifting, with more products relying on a combination of both types of software. It is therefore time for both sides to learn from each other – and particularly for application developers to adopt the debugging best practice that their embedded colleagues have followed over the past twenty years. Otherwise, they face spending ever more time searching for bugs, with consequent impact on productivity, deadlines and company reputation.

Software runs our world – which means debugging must be central to every developer's work. •



YOUR PARTNER FOR QUALITY QUARTZ FREQUENCY CONTROL PRODUCTS, FROM INITIAL DESIGN TO FINAL PRODUCTION

- A fully equipped Design & Test Centre in Munich, where our Design Engineers are available to discuss optimal timing solutions and layout issues for your initial design.
- 3-D images available on request to help with your Design work
- Free Geyer Y-Quartz App analysing tool to help optimize your oscillator circuit and identify the best crystal specifications for your application



All supported by our network of offices in Europe, Asia and America, and factories in Taiwan, S Korea and Japan.



GEYER ELECTRONIC UK Ltd

39 Basepoint, Abbey Park, Romsey, Hants, SO51 9AQ, UK
Tel: +44 (0) 1794 319318
sales@geyer-electronic.co.uk • www.geyer-electronic.com

ODU-MAC®

⊕ Compact modular connector system



VERSATILE MODULAR CONNECTOR SOLUTIONS

ODU MAC – our versatile modular connector solution enables signal, current, data rates, liquid, air and fiber optic transfer.

⊕ COMPLETE SYSTEM SOLUTIONS

ODU provides pre-assembled component solutions. The high level of expertise in development and manufacturing combined with our state-of-the-art manufacturing facilities from Europe, China and USA enable us to offer to our customers tested assemblies and full logistics services.

⊕ ASSEMBLY



ODU-UK Ltd.
Phone: +44 1509 266433
sales@odu-uk.co.uk
www.odu-uk.co.uk



A PERFECT ALLIANCE.

SYSTEM DESIGN EXPLORATION WITH OpenCL FOR FPGAs

DMITRY DENISENKO AND MYKHAILO POPRYAGA FROM ALTERA DESCRIBE THE DESIGN OF A TWO-MILLION-POINT FREQUENCY-DOMAIN FILTER USING THE COMPANY'S OPENCL SOFTWARE DEVELOPMENT KIT FOR FPGAs



Fast Fourier Transform (FFT) is the backbone of signal processing applications. For a long time FPGA vendors have been providing well-tuned FFT libraries to process data sets that fit in FPGA's on-chip memory. But what could be done if the data set is too large?

To solve this problem, the FPGA designer must make multiple intertwined design decisions, such as on-chip FFT core configuration options, their number, how they connect and access external memory, synchronization among multiple cores, and many more. Exploring such design decisions to create the perfect combination for a product – whilst coding in HDL (hardware description language) – is too time-consuming and inconsistent. With higher level programming languages, such as OpenCL (Open Computer Language), system design exploration can be done in days.

This article presents the development of a frequency domain filter supporting between 1M (one million) and 16M points on current FPGA architectures supporting different sample rates from 120 to 240 million samples per second (MSPS). Design decision options for a 2M-point single-precision frequency domain filter using OpenCL will be investigated as an example.

Setting Out The Design

Multi-million-point single-precision frequency domain filters translate inputs into frequency domain using multi-million-point 1D FFT, then multiply each frequency and phase component by a separate user-provided value and translate the results back into time domain with inverse FFT. The complete system's overall target

performance requirement is 150 million points per second for two-million-point sample size on current generation FPGA with two DDR3 external memory banks. Inputs and outputs go directly into the FPGA via 10G Ethernet.

For this design we used Altera's OpenCL SDK (software development kit) for FPGAs compiler targeting BittWare S5-PCIe-HQ

board with Stratix V GSD8 FPGA. We picked OpenCL instead of a lower-level language for two reasons. The first is that designing multi-million-point filters requires building complicated yet highly efficient external memory systems. With lower-level design tools, creating individual blocks such as an on-chip FFT or a corner-turn is relatively easy, especially since every FPGA vendor already provides libraries containing such blocks. However, creating the external memory system would normally require a lot of HDL work. This is especially challenging because you don't know the

“Designing multi-million-point filters requires building complicated, yet highly efficient, external memory systems”

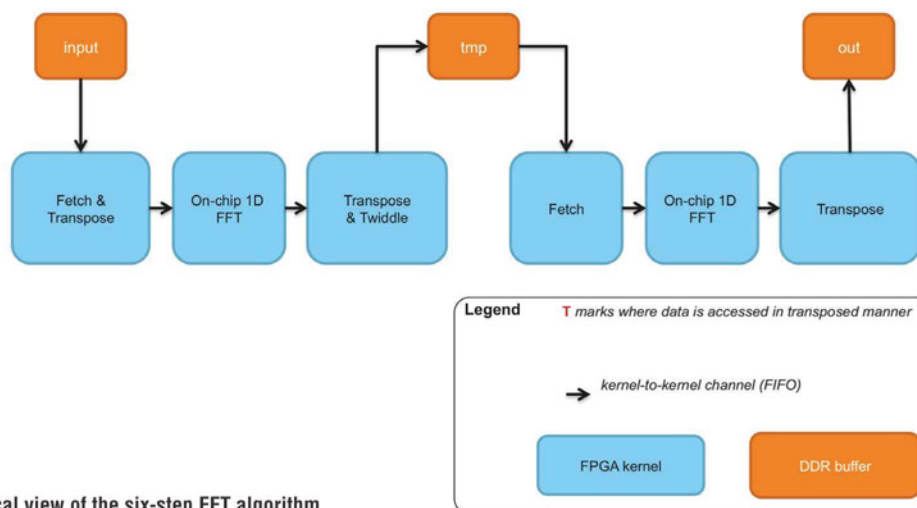
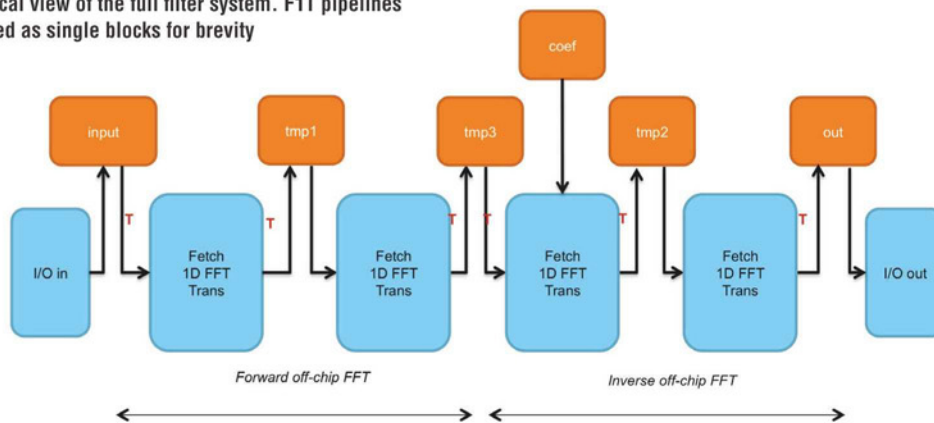


Figure 1: Logical view of the six-step FFT algorithm

Figure 2: Logical view of the full filter system. F1T pipelines are represented as single blocks for brevity



configuration of the overall system when starting.

The second reason for choosing OpenCL is host-level control over the FPGA logic. It was clear from the start that two full copies of multi-million-point FFT cores would not fit on a single device, so a single data set would have to pass over the FPGA logic at least twice before the final output is produced. Coordinating such sharing whilst also allowing features such as dynamically changing data-set size, multiplication coefficients and even completely changing FPGA functionality for something else, is best left to a CPU.

OpenCL compiler for FPGAs solves both these challenges – it builds a customized and highly efficient external memory system and allows fine-grained control over the FPGA logic.

On-Chip FFT

We assume that we already have an FFT core that handles data sizes that fit fully on an FPGA (we will refer to this as “on-chip FFT”), since every FPGA vendor provides such cores. Such core is parameterizable in at least the following ways:

- Data type (fixed or single-precision floating point);
- Number of points to process, N ;
- Number of points to process in parallel, POINTS; and
- Dynamic support for changing the number of points to process.

Given such an on-chip FFT core, building the overall system requires two steps: building an FFT core that can handle multi-million points, and then stitching two such cores together, with complex multiplication in between to create the whole system.

The classic way to implement FFT with external storage is the six-step algorithm that treats a single one-dimensional data-set as two-dimensional ($2M = 2K \times 1K$), as shown in Figure 1, which also shows both the separate computation kernels and external memory buffers.

“Fetch” kernel reads data from external memory, optionally transposing it, and outputs it into a channel (also known as a “pipe” in OpenCL 2.0 nomenclature). In hardware, a channel is implemented as a FIFO with compiler-calculated depth. “On-chip 1D FFT” is unmodified vendor’s FFT core, taking input and

producing bit-reversed output using channels. “Transpose” always transposes the data it reads from its input channel, optionally multiplying it by special twiddle factors, and writing the output in natural order to external memory.

As shown in Figure 1, data is sent over Fetch → 1D FFT → Transpose (F1T) pipeline twice to produce the final output. This gives us our first important design choice – either have one copy of F1T pipeline to save area or two copies for possibly higher throughput.

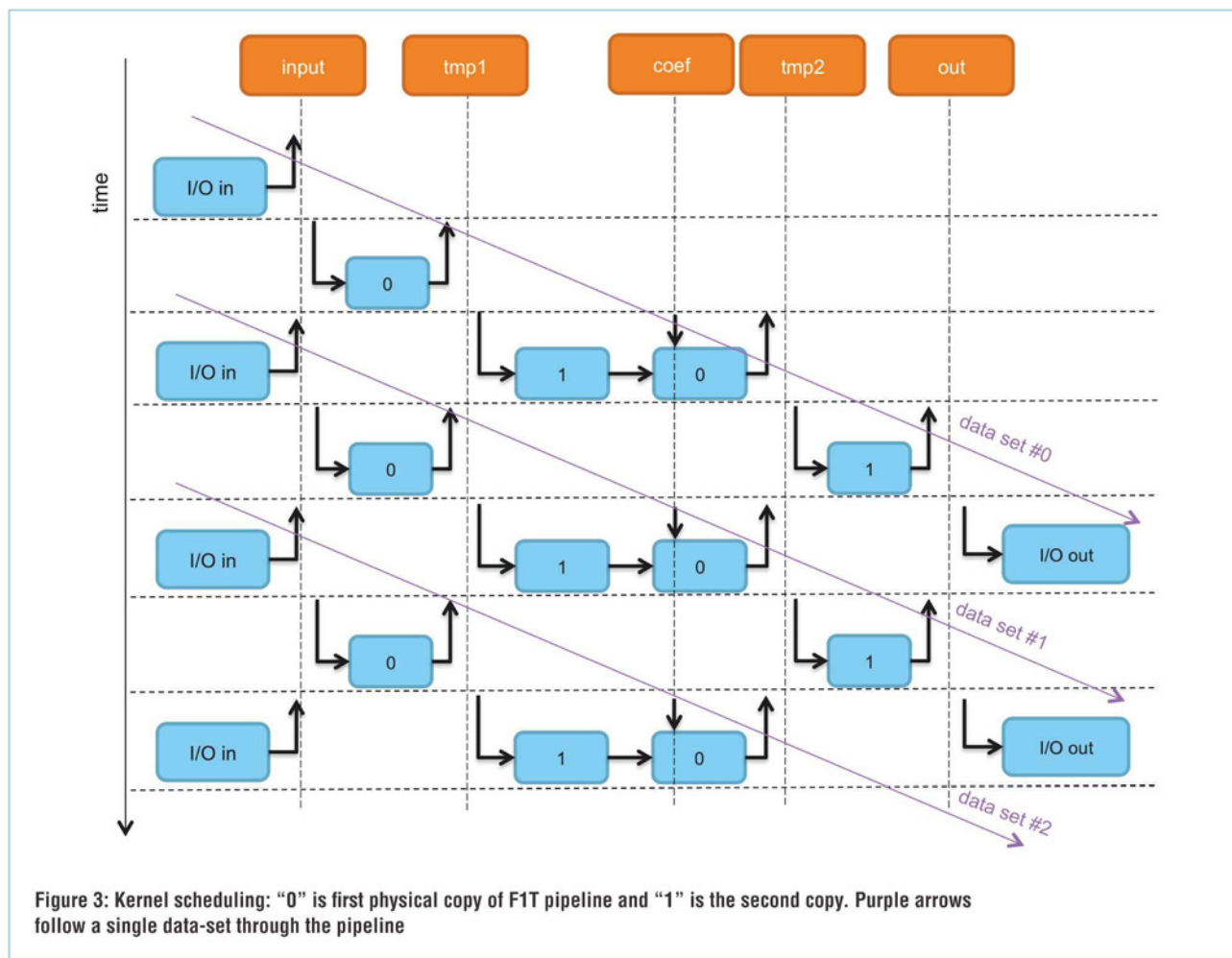
Prototyping The Algorithm

The algorithm was first prototyped in an emulator to get the address manipulation for transpositions and twiddle factors correct. An emulator compiles OpenCL kernels to x86-64 binary that can run on a development machine without an FPGA.

Moving from emulator to hardware compile was a painless step – functionally correct code in an emulator became functionally correct code in hardware, with no simulations necessary. The only aspect we had to modify, for performance and area reasons, was the local memory system used by the Fetch and Transpose kernels.

Efficient transposition requires buffering POINTS columns/rows of data in local memory. The OpenCL compiler analyzes all accesses to local memory in your OpenCL code and creates a custom on-chip memory system optimized for that code. In the case of POINTS=4, our original transposition kernels had four writes and four reads. An on-chip RAM block, double-pumped, can service at most four separate requests, with at the most two being writes. To support four writes and four reads, on-chip memory needs to be both duplicated and contain request arbitration logic, causing area bloat and performance loss.

Once we realized that the write pattern could be changed to make all four writes consecutive, these four writes were grouped by the OpenCL compiler into a single wide write, giving us only five accesses to the local memory system: one write and four reads. With that change, the compiler



automatically built us a much smaller five-ported memory system that could service all five requests on every clock cycle without stalls.

Once the design compiled to hardware, we could do performance measurements. With a single copy of F1T pipeline on the FPGA, we measured 217 MSPS with POINTS=4 and 457MSPS with POINTS=8 for four million point FFT. The POINTS=8 version used twice as many on-chip block RAMs, and two copies in this configuration would not fit. This gave us the first design dimensions to explore – number of points to process in parallel versus area.

Full Filter Design

Now that we have a multi-million point FFT, we are ready to put the whole design together. Simply stitching two off-chip FFTs gives the logical view of the whole pipeline in Figure 2.

Besides duplicating a single off-chip FFT computation pipeline, the following parts were added to the system: complex multiplication in frequency domain is absorbed into third F1T block; the coef buffer is holding two million complex multiplication coefficients.

I/O-in and I/O-out kernels were added to realistically model an additional load on external memory of 10G Ethernet channels. With these kernels, we could continue with a purely software-based development and leave the Ethernet channel integration until after the core computation pipeline was fully optimized. The I/O in the kernel generates a single sample every clock cycle and I/O out consumes a single sample every clock cycle.

As experiments with off-chip FFT showed, we can fit only two F1T blocks and only with POINTS=4. Therefore, the data has to pass through the hardware twice for full computation. That gave us overall system throughput of only 120MSPS for 2M points, below our target of 150MSPS. By reducing the data size to 1M points, we were able to fit POINTS=8 version and get throughput of 198MSPS, which showed that there is still performance to be had, if only we could make the POINTS=8 version fit for 2M points.

Picking an optimized structure of the full pipeline in Figure 2 is the next step in the overall design process. First, we removed the tmp3 buffer. Both sides access it in the same way (transposed write and read); therefore, second and third F1T blocks can be connected directly by a channel. This required making the Transpose kernel either write its output to external memory or into a channel,

and a similar change for Fetch. Such a change is dynamically controlled by the host, so a single physical instance of Fetch could be used, which incidentally changed our connectivity to external memory. However, this is something we did not have to worry about as the OpenCL compiler always generated efficient custom external memory interconnect for our system.

A further improvement would be to move the second transpose “T” from writing to tmp1 to reading from tmp1 (the data in tmp1 would be stored differently but the net effect is the same). This would eliminate the need for one local memory, buffer used by transpositions. Even though this change would not be hard to implement, we decided to forgo it for a more radical idea.

Our original implementation of transposition is done in two stages: first, all the required data is loaded into local memory then it's read from local memory using transposed addresses. To efficiently utilize such pipeline, the OpenCL compiler automatically double-buffers the local memory system. This way, the loading part of the pipeline can load the data into one copy, while the reading part can read previous data-sets from another copy. The automatic double-buffering is the right thing to do for our transposition algorithm but it's expensive. Instead, we rewrote the transposition kernels to be in-place. Such a kernel needs only a single buffer and supports reading and writing

multiple data points at the same time.

With these changes, we were able to fit 2M point FFT with POINTS=8 configuration and achieve 164MSPS throughput.

Scheduling

Only two copies of F1T could fit. Figure 3 shows how we scheduled data flow to fully utilize the pipeline. Notice that in steady state, the pipeline alternates between processing two and three data-sets at the same time without additional buffers. This scheduling is controlled by the host program running on CPU and was verified using the Dynamic Profiler tool.

Buffer Allocation

In an OpenCL system the host program controls which DDR bank contains which buffers. Since a DDR bank is most efficient when it's either read from or written to but not both, we split the five buffers among two DDR banks as follows:

- DDR bank #0 gets input and tmp2.
- DDR bank #1 gets tmp1, coef, and out.

Assigning a buffer to a DDR bank is a one-line change in the OpenCL host program. The compiler and the underlying platform take care of the rest. Given such automation, we experimented on 2-DDR and 4-DDR boards to find the best mapping of buffers to banks for each board. ●

Electronics WORLD

Your essential electronics engineering magazine and technical how-to-guide

A subscription to Electronics World offers



SUBSCRIBE TODAY FROM JUST
£46 BY VISITING THE WEBSITE OR
CALLING +44(0)1635 879 361

www.electronicsworld.co.uk/subscribe

Register for our free newsletter,
please scan here



Electronics World provides technical features on the most important industry areas, including: Nano measurement • Signal processing • Connectors • Semi-conductors • Embedded • Automotive • Power supplies • DSPs • Cables • Microwave • Test & Measurement • Power Supplies • Communications • Lighting • USB Design • RF • Robotics and much more...

OPTIONS FOR PROVIDING IoT WIRELESS CONNECTIVITY

BY **RUI RAMALHO**, PRODUCT MANAGER FOR CONNECTIVITY MODULES AT MURATA EUROPE



any embedded developers are increasingly finding an additional requirement for their designs – every Internet of Things (IoT) application requires wireless connectivity to transfer sensor data to the cloud. As such, engineers can either design their own wireless transceiver or decide to incorporate

a pre-certified wireless module.

First, they need to choose the communications protocol for the design. There are many communications technologies available, with the most popular being Wi-Fi, Bluetooth and ZigBee. Then, the designer needs to investigate the amount of data to be transferred and the range; for most applications, there are always trade-offs between range, data rate and application.

Choosing The Comms Standard

The application will highlight a number of criteria to consider, such as the frequency at which data needs to be transferred and the supply-power budget, especially for battery-powered devices. For example, for controlling a device or receiving sensor data only a few times per day, Bluetooth Low Energy (BLE) can be a suitable option, since it is a standard supported by virtually any smartphone nowadays. However, if a higher quantity of data, say a few Mbytes, needs to be transmitted, then Bluetooth Classic or Wi-Fi might be a more suitable option.

BLE typically communicates over 30m in line-of-sight, but if data needs to be transmitted across a building, then an alternative should be considered, like the emerging technology

of BLE Mesh, ZigBee or sub-1GHz. The range can be extended by data being repeated along the mesh, which requires nodes along the way in the network. This may present challenges when a sensor is situated at one end of a building, for example, and the access to its information is at the other end.

Sub-1GHz ISM (industrial, scientific and medical) band communication technologies may be inconvenient in some cases because they are not using uniform frequencies across the world, thus require different product variants or adjustments for different regions. In addition, the communication protocols used are not standard, creating even more incompatibilities between devices.

For an increasing number of IoT designs, Wi-Fi is seen as a good communication method for several reasons. First, the

infrastructure is becoming more available; second, the data rates are much higher; and, third and most significantly, Wi-Fi usually provides a direct connection to cloud-based services.

Type Of Design

Once the communication protocol has been selected, the engineer needs to choose the components for the design. At this stage, typically the question is “should I use a

“Management will, no doubt, also appreciate that there are no additional resources for the development, such as expensive test equipment and, potentially, at least one specialist wireless communication design engineer

module or a make a discrete design?”.

There are two situations where a module is usually the logical answer: when time-to-market and/or size matter. A module is a ready-made, single, fully-tested component and, with a high integration level, has a much smaller footprint than a discrete design. This is the reason why top smartphone manufacturers choose modules for their designs.

In an industry where cost is very sensitive, often the principal argument is the price of components. Here, the engineer must be aware of the important differences of price versus cost. When the bill of materials (BOM) is compared between a discrete connectivity solution and a module, the module usually costs a bit more. Obviously, there are the assembly costs that need to be considered in addition to the development, test and certification costs. The engineer must have always in mind that a module is

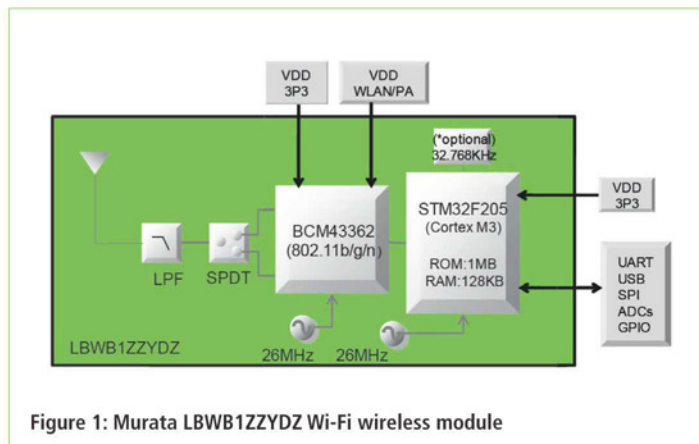


Figure 1: Murata LBWB1ZZYDZ Wi-Fi wireless module

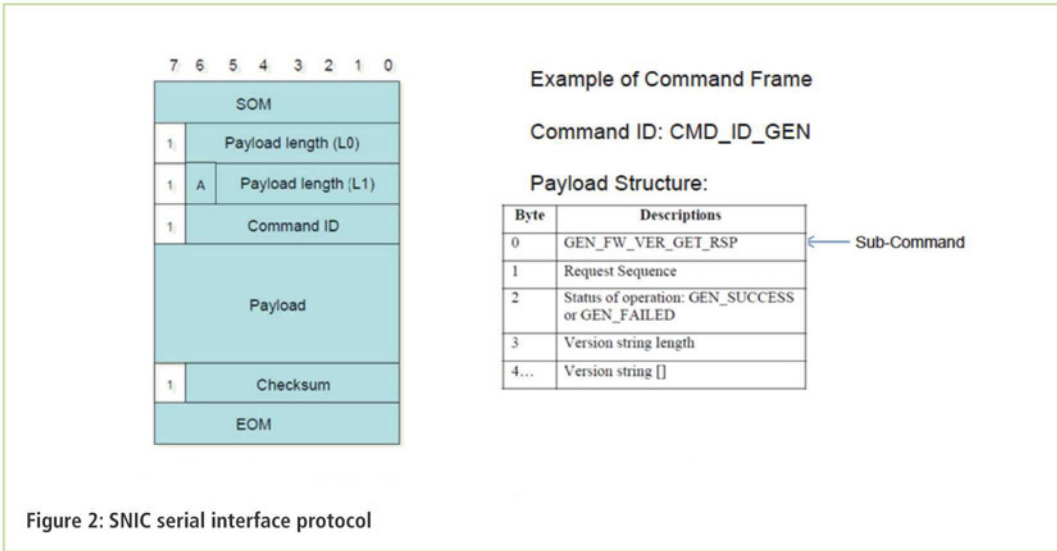


Figure 2: SNIC serial interface protocol

a single component fully tested on the production line, ready to be placed on the PCB and, in most cases, pre-certified by many regional wireless certification bodies.

Modules with pre-integrated firmware are also available to simplify the task of software integration. Procurement colleagues will have fewer individual components to manage in the supply chain, which will guarantee a smooth production. Management will no doubt appreciate there are no additional resources for the development, such as expensive test equipment and, potentially, at least one specialist wireless communication design engineer. Modules also avoid the hidden costs associated with, for example, the expensive re-design of an RF board due to EMI problems caused by poor track layout.

Choosing The Module

Having selected the module route for implementing Wi-Fi connectivity, the engineer can pick from a number of suitable products on the market, such as for example the Type YD (part number LBWB1ZZYDZ) from Murata (see Figure 1).

Measuring just 33 x 18 x 2.5mm, this compact FCC/IC and CE-compliant module integrates a Broadcom BCM43362 wireless transceiver, a STMicroelectronics STM32 ARM Cortex-M3 microcontroller and an antenna. Several software architectures can be supported. Using Murata's Simple Network Interface Controller (SNIC) protocol, which is based on the Broadcom WICED solution, it offers access to a pre-built firmware/software package that makes host integration, wireless and peripheral control much easier, and relieves the burden of firmware programming. Acting like a black box, the module can be controlled by any MCU through UART or SPI interfaces.

Figure 2 illustrates how message exchange is achieved between the customer application and the Type YD Wi-Fi module. Figure 3 shows some of the command instructions issued over the UART or SPI interface to detect Wi-Fi access points, establish communication and transfer data.

SNIC Communication Sub-Command

Sub Command ID	Description
GEN_FW_VER_GET_REQ	Get firmware version string
SNIC_SEND_FROM_SOCKET_REQ	Send from socket
SNIC_CLOSE_SOCKET_REQ	Close socket
SNIC_SEND_ARP_REQ	Send ARP request
SNIC_GET_DHCP_INFO_REQ	Get DHCP info
SNIC_RESOLVE_NAME_REQ	Resolve a host name to IP address
SNIC_IP_CONFIG_REQ	Configure DHCP or static IP
SNIC_DATA_IND_ACK_CONFIG_REQ	ACK configuration for data indications
SNIC_TCP_CREATE_SOCKET_REQ	Create TCP socket
SNIC_TCP_CREATE_CONNECTION_REQ	Create TCP connection server
SNIC_TCP_CONNECT_TO_SERVER_REQ	Connect to TCP server
SNIC_UDP_CREATE_SOCKET_REQ	Create UDP socket
SNIC_UDP_START_RECV_REQ	Start UDP receive on socket
SNIC_UDP_SIMPLE_SEND_REQ	Send UDP packet
SNIC_UDP_SEND_FROM_SOCKET_REQ	Send UDP packet from socket

Figure 3: Typical solar cell circuit

Figure 3: Command set examples

It should also be noted that, in addition to using the module as a communications interface, an engineer can do a complete application development on the module's internal ARM Cortex-M3. This would be ideal for a number of applications that involve connecting a small number of sensors, for example to a cloud application without the need of a host application MCU. There are a number of unused GPIO pins available on the module to allow this in addition to the MCU's ADCs. The final firmware can then be downloaded to the module such that it becomes the core of the device.

Selecting a pre-certified wireless module instead of developing a discrete solution will save significant amount of design resources and cost, but will also ensure you can get your product to market as quickly as possible. ●

TIME TO TAKE THE HEAT OUT OF LEDs

USING ARM A9 CORES ON A XILINX ZYNQ SYSTEM-ON-A-CHIP CAN SIGNIFICANTLY INCREASE THE PERFORMANCE OF A SYSTEM, WRITES **ADAM P. TAYLOR**, CHIEF ENGINEER FOR ELECTRICAL SYSTEMS AT E2V

One of the many benefits of the Xilinx Zynq-7000 all programmable system-on-a-chip (SoC) is that it has two ARM Cortex-A9 processors on board. However, many bare-metal applications and simpler operating systems use only one of the two ARM cores in the Zynq SoC's processing system (PS), a design choice that can potentially limit system performance.

Depending on the application, there could be a need to have both processors running bare-metal applications, or to run a different operating systems on each processor. For instance, one side could perform critical calculations on a bare-metal/RTOS application, while the second processor provides HMI and communications using Linux.

About Multiprocessing

Either scenario is an example of multiprocessing, which briefly defined is the use of more than one processor within a system. A multiprocessing architecture can allow the execution of multiple instructions at a time, although it does not necessarily have to.

There are two kinds of multicore processing: symmetric and asymmetric. Symmetric multiprocessing makes it possible to run a number of software tasks concurrently by distributing the load across a number of cores. Asymmetric multiprocessing (AMP) uses specialized processors or applications execution on identical processors for specific applications or tasks. Using both cores on the Zynq SoC with bare metal or different operating systems is by definition an example of asymmetric multiprocessing.

AMP on the Zynq SoC can involve any of the following combinations:

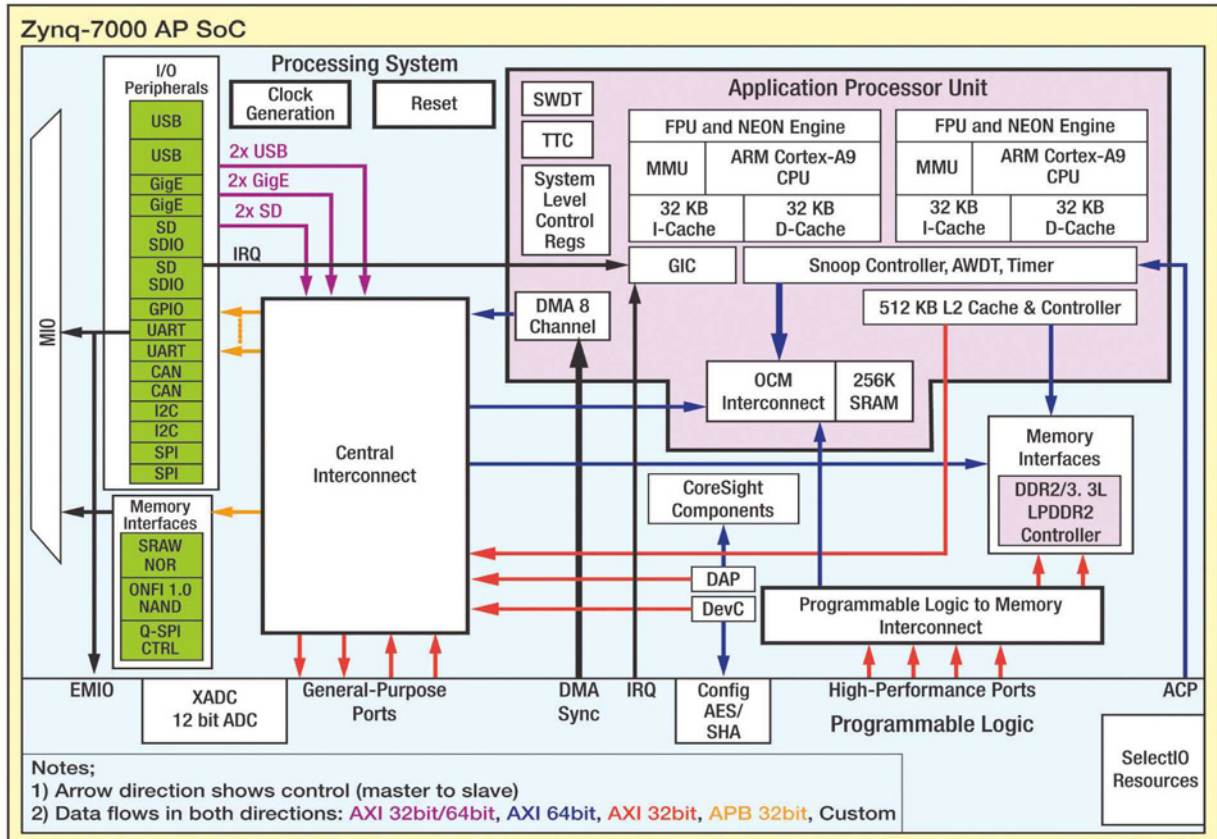


Figure 1: The Zynq SoC processing system, showing private and shared resources

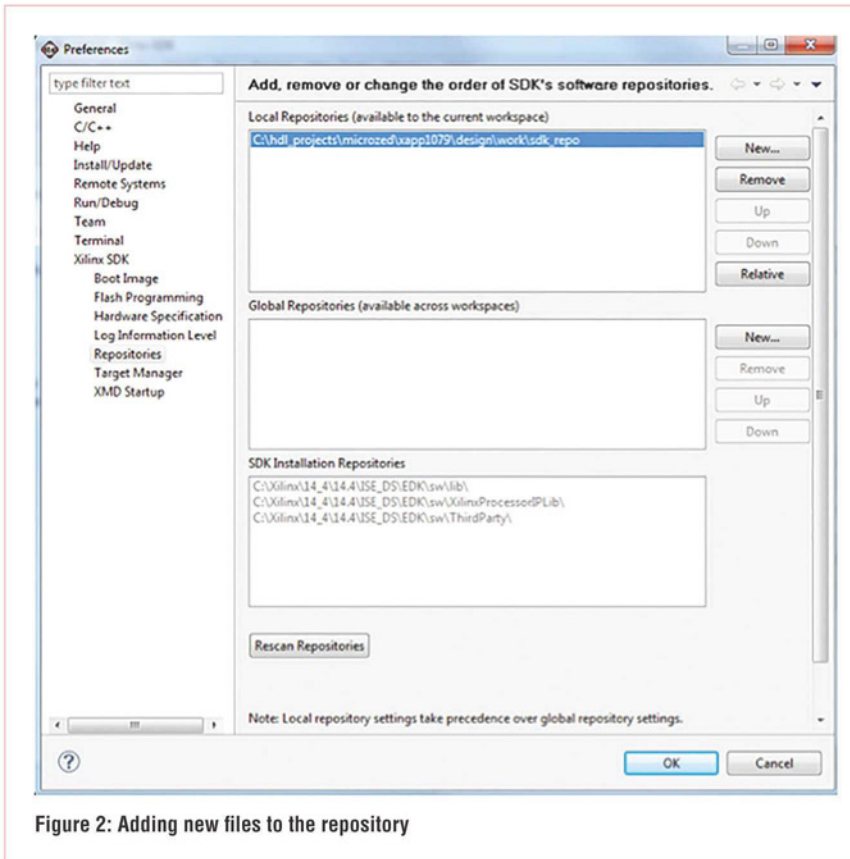


Figure 2: Adding new files to the repository

- Different operating systems on Core 0 and Core 1.
- Operating system on Core 0, bare metal on Core 1 (or vice versa).
- Bare metal on both cores executing different programs.

When creating an AMP system on the Zynq SoC, it has to be taken into account that the ARM processor cores contain a mixture of both private and shared resources that must be correctly addressed and managed. Both processors have private L1 instruction and data caches, timers, watchdogs and interrupt controllers (with both shared and private interrupts). A number of shared resources also exist, of which common examples include I/O peripherals, on-chip memory, interrupt controller distributor, L2 cache and system memory located within the DDR memory (see Figure 1).

Each PS core has its own interrupt controller and is capable of interrupting itself, with one or both cores using software interrupts. These interrupts are distributed by the ARM's Distributed Interrupt Controller.

Since the program for each core is located within DDR memory, great care has to be taken to ensure that the applications have been correctly segmented.

Each PS core has its own interrupt controller and is capable of interrupting itself, with one or both cores using software interrupts

Getting AMPed Up

The key aspect in getting AMP up and running on the Zynq SoC is a boot loader that will look for a second executable file after loading the first application into memory. Xilinx offers an application note and source code in XAPP1079, which comes with a modified first-stage boot loader (FSBL) and modified standalone OS, to create an AMP system.

First, the ZIP file that comes with this application note needs to be downloaded before extracting the FSBL and OS to the desired working directory. Then, the folder called SRC has to be renamed "design". Here, it's important to make sure the software development kit (SDK) is aware of the existence of these new files

containing both a modified FSBL and a modified standalone OS, so the next step would be to update the SDK repository to make it aware of their existence.

This is straightforward: within SDK under the Xilinx tools menu, select "repositories" and then "new", navigating to the directory location <your working directory>\app1079\design\work\sdk_repo as shown in Figure 2.

Communicating Between Processors

Before creating the applications for the AMP design, you'll need to consider the way the applications will communicate (if they need to). The simplest method is to use the on-chip memory. The Zynq SoC has 256kB of on-chip SRAM accessible from the following sources:

- Either core via the snoop control unit (SCU);
- Programmable logic using the AXI accelerator coherency port (ACP) via the SCU;
- Programmable logic using the high-performance AXI port via the on-chip memory (OCM) interconnect;
- Central interconnect, again via the OCM.

Each of these sources can read and write the on-chip memory, so it is especially important to understand the operation of the OCM in detail before using it.

Since there are multiple sources accessing the OCM, it is sensible to define arbitration and priority. As the lowest latency

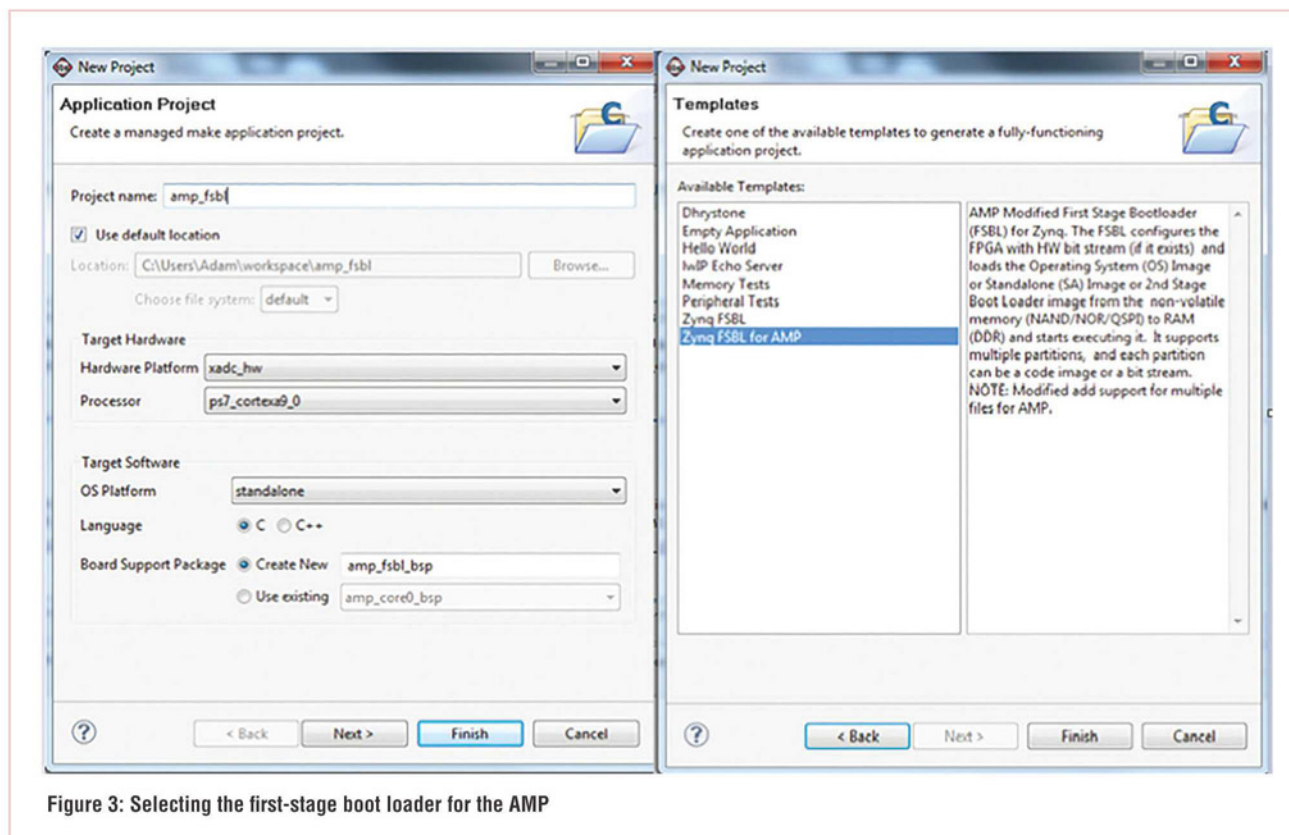


Figure 3: Selecting the first-stage boot loader for the AMP

is required by the snoop control unit, which is either a processor core or an AXI ACP interface, an SCU read from these sources has the highest priority, followed by the SCU write and then the OCM interconnect read and write. The user can invert the priority between the SCU write and the OCM interconnect access by setting the SCU write priority low in the on-chip memory control register.

The OCM itself is organized as 128-bit words, split into four 64kB regions at different locations within the PS address space. The initial configuration has the first three 64kB blocks arranged at the start of the address space and the last 64kB block toward the end of the address space (see Figure 5).

A Simple On-Chip Memory Example

The OCM can be accessed using Xilinx I/O functions to read and write to and from the selected memory address. These functions, which are contained within `Xil_IO.h`, allow for storing and accessing 8-, 16- or 32-bit char, short or int within the CPU address space. Using these functions just requires the address you wish to access and the value you wish to store there. If it is a write, for example:

```
Xil_Out8(0xFFFF0000,0x55);read_char = Xil_In8(0xFFFF0000);
```

A better way to use this technique is to have a common

header file, to ensure the addresses are both targeting the same location within the on-chip memory, especially if different people are working on the different core programs. This file will contain macro definitions of the address of interest for that particular transfer, for instance:

```
#define LED_PAT 0xFFFF0000
```

An alternative approach is for both programs to access the memory location using a pointer. This can be done by defining the pointer, which points to a constant address, normally in C, using a macro:

```
#define LED_OP (*(volatile  
unsigned int *)0xFFFF0000)
```

Again, another macro definition may be used for the address to ensure that the address is common to both application programs. This approach does not then require using the Xilinx I/O libraries and instead allows simple access via the pointer.

Interprocessor Interrupts

The Zynq SoC has 16 software-generated interrupts for each core. As mentioned earlier, each can interrupt itself, the other core or both cores. Using software interrupts is not too different from using hardware interrupts except, of course, in the way they are triggered.

The use of software interrupts frees the receiving application from having to poll an expected memory location for an updated value.

Within both cores, the generic interrupt controller needs to be configured, just as with any hardware interrupt (see Xcell Journal issue 87, “How to Use Interrupts on the Zynq SoC”, for further information).

A software interrupt can be triggered by updating the core using the `XScuGic_SoftwareIntr` function provided within `xsugic.h`. This command will issue a software interrupt to the identified core, which can then take the appropriate action:

```
XScuGic_SoftwareIntr(<GIC
Instance Ptr>, <SW Interrupt ID>, <CPU Mask>)
```

Creating Applications

Having updated the repositories, the next stage is to generate three crucial pieces of the AMP solution: the AMP first-stage boot loader, the Core 0 application and the Core 1 application. Each requires a different board support package to be generated.

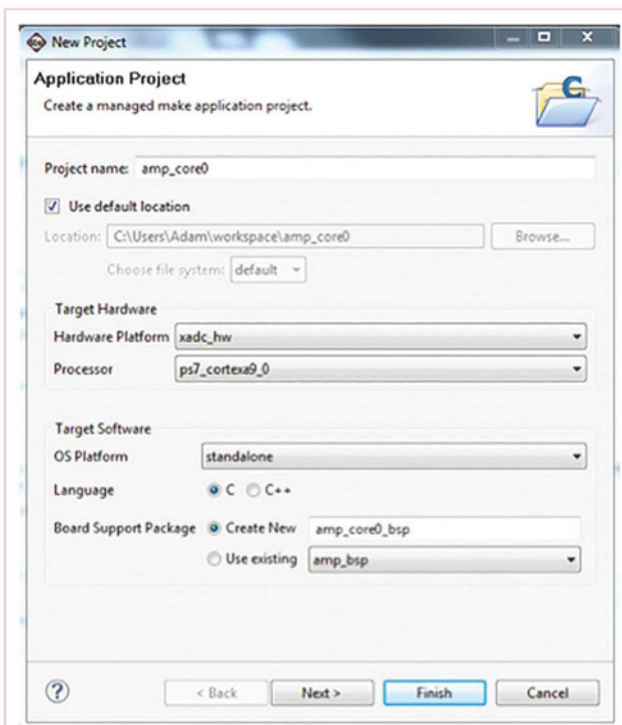


Figure 4: Creating the application and BSP for Core 0

First, create a new FSBL with the SDK. Selecting “file new application project” creates a FSBL project that supports AMP, which is not that different from creating a normal FSBL. However, this time the “Zynq FSBL for AMP” template has to be selected, as shown in Figure 3.

Following the creation of the AMP FSBL, creating the application for the first core is next. Be sure to select Core 0 and your preferred operating system, and allow it to create its own BSP, as shown in Figure 4.

Having created the application, you’ll need to correctly define the location, within DDR memory, from which it will execute. To do this, edit the linker script as in Figure 5 to show the DDR base address and size. This is important, because if the DDR memory for Core 0 and Core 1 applications is not correctly segmented, there’s a risk of one corrupting the other.

Following the segmentation, write the application to be executed on Core 0, since this is the core in charge within the AMP system. Core 0 must start the execution of the Core 1 application; this section of code has to be included within the application (see Figure 6). This code disables the cache on the on-chip memory and writes the start address of the Core 1 program to an address that Core 1 will access. Once Core 0 executes the Set Event (SEV) command, Core 1 will start executing its program.

The next step is to create a BSP for Core 1. It’s important to use the modified standalone OS (standalone_amp), shown in Figure 7, which prevents reinitialization of the PS snoop control unit. As such, do not allow automatic generation of the BSP while you create the project as with Core 0. Be sure to select Core 1 in the CPU selection options.

Now that the BSP for Core 1 has been created, you’ll need to modify the settings of the BSP before moving on to create the application on Core 1. Doing so is very simple and requires adding an extra compiler flag of `-DUSE_AMP=1` to the configuration for the drivers section of the BSP.

With this step complete, the application for Core 1 can be written. Be sure to select Core 1 as the processor and use the BSP you’ve just created. Again, having formed the new application, you need to once more define the correct memory locations within the DDR memory from which the Core 1 program will execute. This is achieved by editing the linker script for the application for Core 1, as before. As with the first core, within this application you must disable the cache on the on-chip memory, which you can use to communicate between the two processors.

Available Memory Regions

Name	Base Address	Size
ps7_ddr_0_S_AXI_BASEADDR	0x00100000	0x00100000
ps7_ram_0_S_AXI_BASEADDR	0x00000000	0x00030000
ps7_ram_1_S_AXI_BASEADDR	0xFFFF0000	0x0000FE00

Figure 5: Core 0
DDR location and
size

```
#include <stdio.h>
#include "xil_io.h"
#include "xil_mmu.h"
#include "xil_exception.h"
#include "xpseudo_asm.h"
#include "xscugic.h"

#define sev() __asm__("sev")
#define CPU1STARTADR 0xffffffff
#define COMM_VAL (*(volatile unsigned long *) (0xFFFF0000))

int main()
{
    //Disable cache on OCM
    Xil_SetTlbAttributes(0xFFFF0000, 0x14de2); // s=b1 TEX=b100 AP=b11, Domain=b1111, C=b0, B=b0
    Xil_Out32(CPU1STARTADR, 0x00200000);
    dmb(); //waits until write has finished
    sev();
}
```

Figure 6: Coding to disable cache on the on-chip memory

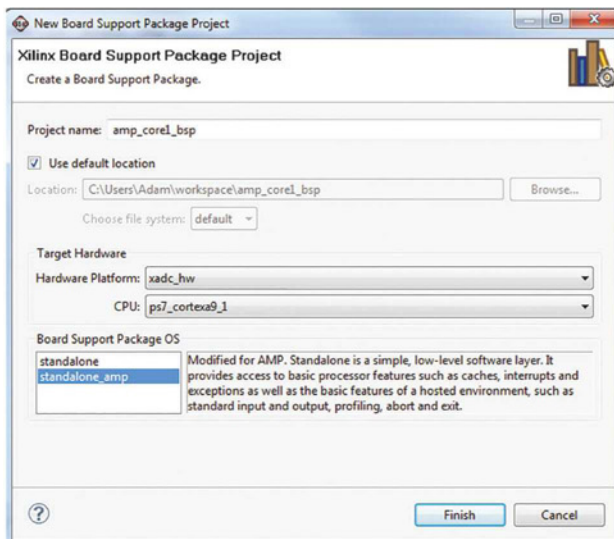


Figure 7: Creating the BSP for Core 1

```
the_ROM_image
{
    [bootloader] amp_fsbl.elf
                download.bit
                amp_cpu0.elf
                app_cpul.elf

    //write start vector address 0xFFFFFFFF0 with 0xFFFFFFFF00
    //This load address triggers fsbl to continue
    [load = 0xFFFFFFFF0] cpu1_bootvec.bin
}
```

Figure 8: Modifying the BIF file

```
C:\Xilinx\14.4\ISE_DS>cd C:\hdl_projects\microzed\xapp1079\design\work\bootgen
C:\hdl_projects\microzed\xapp1079\design\work\bootgen>createboot.bat
C:\hdl_projects\microzed\xapp1079\design\work\bootgen>bootgen -image bootimage.h
if -o 1 BOOT.BIN -w on
C:\hdl_projects\microzed\xapp1079\design\work\bootgen>
```

Figure 9: The creation of the boot.bin file to run on the Zynq SoC

Putting It All Together

Once you have created the applications and built the projects, you should now be in possession of the following components:

- AMP FSBL ELF
- Core 0 ELF
- CORE 1 ELF
- BIT file defining the configuration of the Zynq device upon which you wish to implement AMP.

To enable the Zynq SoC to boot from the selected configuration memory, you will need a .bin file. To create it, you will also need a BIF file, which defines the files to be used to create this BIN file and the order in which they go. Rather than use the “create Zynq” boot image within the SDK, you will be using an ISE Design Suite command prompt and BAT file provided as part of XAPP1079 under the downloaded directory\design\work\bootgen. This directory contains a BIF file and a cpu1_bootvec.bin, used as part

of the modified FSBL to stop it looking for more applications to load.

To generate the BIN file, copy the three generated ELF files to the bootgen directory and edit the BIF file to ensure the ELF names within it are correct, as shown in Figure 8.

Now you can open an ISE command prompt and navigate to the bootgen directory. There, you should run the createboot.bat, a step that will create the boot.bin file, as shown in Figure 9.

This file needs to be downloaded into the nonvolatile memory of the Zynq SoC. Booting the device will result in both cores starting and executing their respective programs.

Creating an asymmetric multiprocessing application on the Zynq SoC can be very simple, using the tools available. It's easy to achieve communication between the two cores using the on-chip memory or even a segmented DDR area. ●

GAN SYSTEMS APPOINTS KRUVAND ASSOCIATES IN THE US

GaN Systems, a developer of gallium nitride power switching semiconductors, has appointed Kruvand Associates as its representative in the US states of Texas, Oklahoma, Arkansas and Louisiana.

Specialist sales engineering and marketing company Kruvand has expertise in power, interconnect, thermal, electromechanical, RF and wireless active and passive components. As sales of GaN power transistors ramp up and replace silicon semiconductors in power applications, Kruvand's Inside Sales and Customer Team will provide pre-sales and design services to GaN Systems's customers and distributor partners.

"As worldwide demand grows for our industry-leading gallium nitride power devices, we are building a network of the very best distributors and representatives. Kruvand Associates is long-established and has an unrivalled reputation as a representative in the West South Central states," said Julian Styles, Director Sales and Marketing, Americas at GaN Systems.

www.gansystems.com



TTI EUROPE AWARDED WITH "2014 PASSIVES DISTRIBUTOR OF THE YEAR" AWARD FROM VISHAY

TTI Inc, a specialist distributor of passive, connector, electromechanical and discrete components, has been named "2014 Passives Distributor of the Year" by Vishay Intertechnology. The prestigious award recognizes outstanding performance using the criteria of sales growth, demand creation, field operation and inventory management throughout the year.

On presenting the award, Philippe Masson, Senior Director Sales Distribution Europe at Vishay said: "We are very proud to present TTI Europe with our '2014 Passives Distributor of the Year' award for recording a strong sales growth for our product line in 2014 and also for performing very well in all the other categories judged. We congratulate the whole team at TTI for their outstanding performance and want to say thank you for your great service and support. We are looking forward to a very fruitful business partnership in 2015."

www.ttieurope.com



CHARCROFT APPOINTS NEW DIRECTORS

Charcroft Electronics, a specialist distributor and manufacturer of components for harsh and high-end applications, announces that Debbie Rowland and Ian Ford have been promoted to Directors of Charcroft Electronics Ltd.

Debbie Rowland has been with Charcroft for over 22 years, initially joining the company as a sales assistant. After a number of successful subsequent roles she was promoted to Sales Manager in 2005. Ian Ford joined Charcroft in 2010 from TT Electronics European Components Group, and initially worked as Charcroft's Purchasing Manager before being promoted to Commercial Manager in 2014.

"The promotion of Debbie and Ian to the Board of Directors will ensure that we continue to maintain our focus on specialist products and support for OEMs and CEMs working in harsh and high-end applications," said Paul Newman, Managing Director.

www.charcroft.com



CONGATEC EXPANDS EXECUTIVE BOARD TO ACCELERATE GROWTH

Congatec, embedded computer boards and modules provider, has expanded its executive board to accelerate company growth and named Jason Carlson as the new CEO, who will be based in the United States.

"The appointment of Jason Carlson fulfills my desire to bring in a new CEO, preferably American, with the experience we need to fulfill our next growth milestones. This allows me to focus on the CTO role within the executive board and enables me to continue to provide the leadership for Congatec's strategic and technological development," said Gerhard Edi, founding member and CEO of Congatec.

Carlson brings over 25 years of experience in leading private and publicly held technology companies. He said: "My mission is to identify and coordinate internal and external growth opportunities, with the goal of elevating Congatec to becoming a global player."

www.congatec.com



MOUSER SPONSORS STUDENT SOLAR CAR TEAM IN UPCOMING CHALLENGE

Mouser Electronics is sponsoring the Ben Barber high school solar racing team of Mansfield, Texas, as it competes in the 2015 Solar Car Challenge. The event will be July 18-23 at Texas Motor Speedway. This is the fourth year in a row that Mouser sponsors the team.

The Solar Car Challenge was established in 1993 to motivate students in science and engineering and increase alternative energy awareness. The challenge teaches high-school students around the world how to build roadworthy solar cars.

"Supporting and encouraging the engineers of tomorrow is a large part of our mission at Mouser," said Kevin Hess, Mouser's Vice President of Technical Marketing. "We are very proud to again support our local Ben Barber solar racing team. It is exciting for us to have a role in supporting this important team project that encourages STEM education."

www.mouser.com



DIGI-KEY AND ARM OFFER 'LAB-IN-A-BOX' FOR UNIVERSITIES

Digi-Key Electronics announced its partnership with the ARM University Program to distribute the innovative 'Lab-in-a-Box' (LiB) to higher educational institutions around the globe.

The LiB contains ARM-based technology and high quality, rigorous training materials that support electronics and computer engineering courses. Since its launch in February 2014, ARM LiBs have been successfully deployed in hundreds of universities worldwide, enabling easy migration for academics wanting to upgrade their existing curricula to state-of-the-art technologies from the vast ARM eco-system.

The Lab-in-a-Box package includes hardware development boards, professional software licenses from ARM, and complete teaching materials from the ARM University Program ready to be deployed in classes, such as lecture notes, slides, demonstration codes, lab manuals and projects with solutions in source.

www.digikey.com



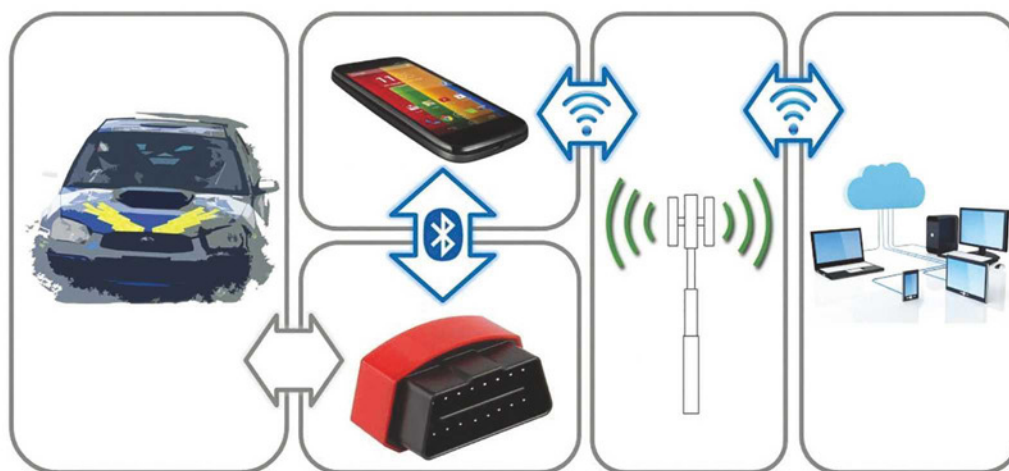


Figure 1: Amalgamation of external and internal data sources feeds the cloud-based video and data servers for user viewing and analysis

FORMULA 1 IN YOUR POCKET: TELEMETRY FOR EVERYONE?

BY TONY DEWHURST, TRACKSPY DEVELOPMENT LEAD

It's fascinating to see how ordinary motorists and amateur/semi-pro race drivers are gripped with obtaining Formula-1-like data and telemetry from their driving activities. So much so that there are a number of very good smartphone apps for the major platforms aiming to provide motorsports telemetry experience to a wide audience. Some are free, some quite expensive, but all have a large following and take advantage of the immense processing power and resources available on modern smartphones, making maximum use of flexibility and connectivity you would be hard pressed to find on some mid-range laptops.

Some of the current crop of smartphone-based motorsports data acquisition and review applications are better implemented than others, but there's a growing trend for this type of application to be complemented by back-end server support for more comprehensive playback and data analysis. So far, such server-based analysis systems are by no means as professional or as comprehensive as the top-end products on offer from the likes of Magneti Marelli (WINTAX), McLaren (ATLAS) or the GEMS (GDA) tools for example, but much more accomplished than those on a smartphone.

A relative newcomer to this arena is the TrackSpy application, currently only available as a preview/beta on the Android/PlayStore platform. It has a back-end server that provides more F1-style analytics than other similar, smartphone-based offerings, incorporating video and data acquisition, as well as competitive and social features. It connects to external data sources, notably the second-generation on-board diagnostics port (OBD-II) of the vehicle's electronic control unit (ECU), a

mandatory fitment on all vehicles manufactured and sold in the EU since 2001 (1996 in the US).

System's Perspective

The development of TrackSpy was tackled from the ground up; it was designed to act as a two-part system to provide intuitive setup for data acquisition (smartphone end), seamless session management and transfer of recorded data to a dedicated video render and display server, and a clear and simple user interface.

Paramount requirement is to allow the driver to clearly establish that all data sources are working properly before recording begins. Among these sources are the vehicle ECU and the smartphone sensors, which include the built-in camera (video sensor), accelerometers ('g' sensors), and gyroscope and location sensors (GPS data).

There is nothing more frustrating than to put in a good stage time or a blistering lap, only to find at the end that half of the data is missing due to a simple oversight, so it was critical that the driver (or co-driver, in the case of stage rallies) can clearly see that the data of interest is ready to go.

For an end-to-end experience, the data recorded by the device needs to find its way to the video and data display servers in the cloud. To this end, industry standard cloud storage and processing resources are employed in the form of AWS cloud services (see Figure 1).

The Android Development Environment

Initially tackled for the Android OS, all the software was written in Java. Although C-like in many of its structures and constructs, Java is essentially an object-oriented language. The Android OS

architecture is based on Linux, and the API support for developers implements a large subset of the Java runtime language, supported by the Dalvik virtual machine.

Although many regard smartphones as computers in their own right, they are essentially resource-limited devices, so the Android flavour of Linux can be considered a small-footprint embedded software system. The TrackSpy application borrows many queues from microcontroller-based race engine and chassis-control systems, but these truly real-time embedded systems run dedicated interrupt-driven software, crafted specifically for the complex task of running high-revving engines in some of the harshest of operating environments.

One of the overriding guiding principles of smartphone development is that the user should never be aware of background processes or long-running tasks, lest they hold up the user interface (UI). Users expect fast-responding applications and the Android OS will force an untimely exit of the application if it doesn't visit the UI for more than a few seconds.

In spite of some flavours of Linux being termed real-time (RT-Linux, for example), it is true to say that no process-oriented, thread-based architecture can ever properly implement a real-time embedded system. "Fast" is not the same as "real-time", and with thread latencies running into several tens of milliseconds, a control system with this level of latency would soon leave you with a gearbox full of false-neutrals and your engine with a hole in the side. Android is no exception here, but where Linux falls short of real-time capability, the speed and performance of the (mainly) ARM-based processor cores used in most Android devices more than compensate, and most Android-based smartphones are therefore more than capable of implementing this non-control type of data acquisition and telemetry system, which truly offers F1 telemetry in your pocket.

The Lifecycle Of An Android Application

Although smartphones are making performance gains with every new generation, the phone is still essentially a resource-limited system and, as such, the Android OS must make sure that best use is made of those limited resources so the user can undertake a multitude of unrelated tasks simultaneously (SMS messaging, emails, taking phone calls, for example) whilst using TrackSpy. Android achieves this non-trivial task by maintaining strict lifecycle control of every non-system application. As shown in Figure 2, this typically consists of an onCreate() callback at the very start of the application's life, an onDestroy() callback at its very end, and a number of optionally-implemented intermediate callbacks that allow the developer to maintain the integrity of the application's data and control space, whilst accounting for the inevitable interruptions that a phonecall or screen shutdown will force.

Minimising battery usage is fundamental to a well-designed Android application. TrackSpy makes full use of intermediate lifecycle callbacks in order to switch off all unnecessary battery drains while the app is not in the foreground of user activity. This ensures that the GPS, accelerometer and Bluetooth radio

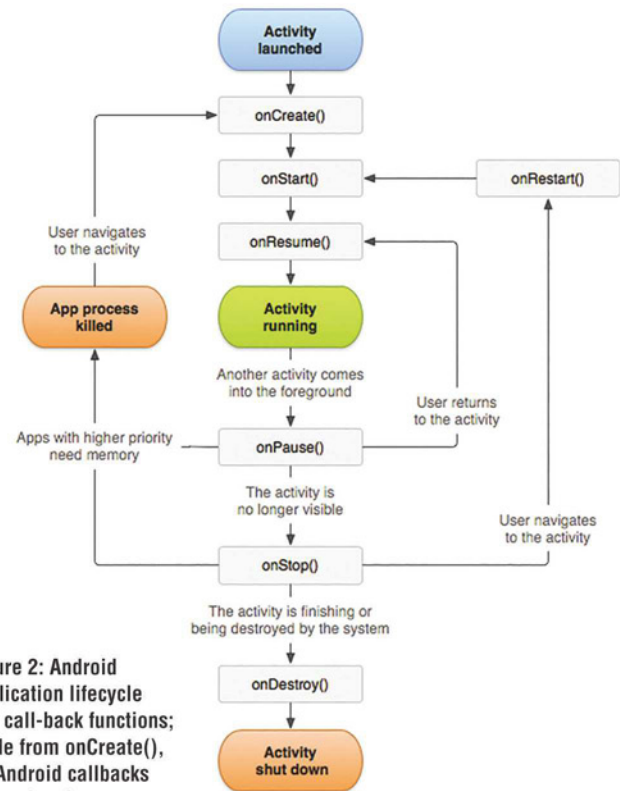


Figure 2: Android application lifecycle and call-back functions; aside from onCreate(), all Android callbacks are optional

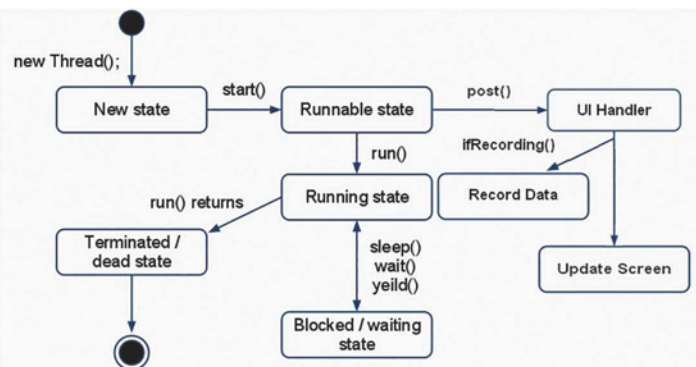


Figure 3: The TrackSpy Update Thread maintains periodic UI updates and data feeds to the logging function

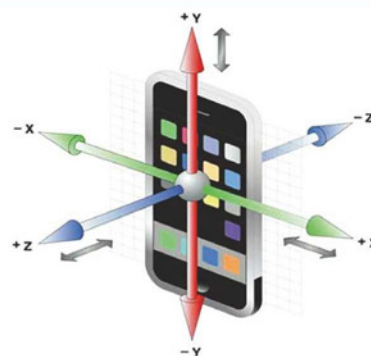


Figure 4: Tri-axial accelerometer axes in Android smartphones

```

/* Calibrate the sensors */
if (true == SetAccelsReq)
{
    byte help_1 = -1, help_2 = -1;

    values_a[0] = TrackSpyActivity.MultDecimal(event.values[0]/g, 100);
    values_a[1] = TrackSpyActivity.MultDecimal(event.values[1]/g, 100);
    values_a[2] = TrackSpyActivity.MultDecimal(event.values[2]/g, 100);

    /* Get position right now */
    for (byte byAxis = 0; byAxis < 3; byAxis++)
    {
        if ((values_a[byAxis] > 70) || (values_a[byAxis] < -70))
        {
            /* **This** axis is more than half way rotated
            ** (there will only ever be one!)
            */
            if (values_a[byAxis] < -70)
            {
                /* Axis is more than half way rotated 'backward' */
                rotate = true;
            }
            else
            {
                /* Axis is more than half way rotated 'forward' */
                rotate = false;
            }

            /* Align the axis */
            byZIndex = byAxis;

            for (byte i_help = 0; i_help < 3; i_help++)
            {
                if (i_help != byAxis)
                {
                    if (help_1 > -1)
                    {
                        help_2 = i_help;
                    }
                    else
                    {
                        help_1 = i_help;
                    }
                }
            }

            byXIndex = help_1;
            byYIndex = help_2;
        }
    }

    SetAccelsReq = false;
}

```

Figure 5: Simple algorithm employed in the TrackSpy application determines the X, Y and Z tri-axial accelerometer sensor indices



Figure 6: TrackSpy data and video view (the video image is obscured)

subsystems are put to sleep while data is not visible to the user (if the user can't see the data, there is no need for it to be acquired for display). This does not hold true for the recording function of course, where interruptions to any of the data sources would lead to gaps in the final recorded session.

Listeners And Threaded Tasks

To keep TrackSpy responsive, whilst ensuring that sensor data gathers and screen updates are maintained at an acceptable rate, judicious use was made of the multithreading and process-oriented features of the Android flavour of Linux.

Some of the data of interest needs to be correctly set up and initialised before the information is available for display and recording. As data is gathered from a number of different sources, it was necessary to design the application to collect it in a timeframe that allowed updates to be fed to the screen fast enough to show that the data source is correctly configured, but not so fast that the inevitable granularity issues blur the screen, nor excessive processor MIPS are consumed.

To receive data from the various sensor device-drivers in Android, the application must register itself to receive updates from the chosen sensor source. This essentially forces the hardware devices behind the data to remain active whilst transmitting regular data updates to the application's listener functions. As such, it is vital these listeners are disconnected when the app goes to the background or shuts down, otherwise this hardware will

remain active, transmitting data into the void.

In addition to the OS callbacks, two main threads are employed to achieve a responsive UI and gather the necessary sensor data in a timely manner: the Update thread and the Sensor thread.

The Sensor thread is just that; once the application is initialised, it registers itself for the GPS, accelerometer and gyro data feeds and sends the data to the screen for the user to check that each of the required sources is reading it. The in-vehicle data read from the OBD-II port in particular needs to be read at a rate that can be supported by the ECU. Some ECUs, notably those in a number of Japanese models, fail to conform to the OBD standard, and if data is requested from them too quickly, they will shut down the data feed.

The Update thread serves two roles: it maintains the responsiveness of the UI and it passes sensor data to the logging function. It keeps the UI responsive by scheduling the acquired data feed to the screen at a high enough rate to allow user input (screen touch, or device rotation), but also to allow other threads to carry out the important work of gathering that data. The Update thread passes sensor data to the logging function so that the race session can be recorded for later analysis (see Figure 3).

Accelerometer Calibration

Uniquely for this type of application, viewing and recording is possible in both portrait and landscape orientation, so the tri-axial accelerometer inputs must be calibrated to compensate for the orientation of the phone before recording starts, otherwise the

X, Y and Z axes will not compare across recorded sessions if recordings are carried out in different orientations.

The tri-axial X, Y and Z accelerometer device axes are set the same in all Android phones, as shown in Figure 4.

In the scenario where the driver uses the device in portrait mode for one run and then landscape mode for a subsequent run, assuming the rear-facing camera was pointing in the direction of travel in all cases, it is easy to see that, although the acceleration and braking forces (the Z axis) would remain the same, corner and bump acceleration data would be transposed and it would be difficult (but not impossible) to compare the two runs. There is certainly the possibility of confusion.

To counter this, the app calibrates the accelerometer sensor inputs at the onset of recording so that the X, Y and Z sensor axes are always consistent with Earth's gravitational pull and not fixed with the phone itself. The function employed to do this essentially works out the orientation of the smartphone based on the raw accelerometer data, to transpose the X, Y and Z inputs of the phone into the X, Y and Z axes based on which sensor axis is most affected by gravity pull. Figure 5 shows the simple algorithm developed for TrackSpy to do this.

Following the call to the calibration algorithm, the acceleration sensor's data-array indices byXIndex, byYIndex and byZIndex are set to the array elements aligned with the Earth's gravitational pull. The net result is that the X axis always measures 'bump', the Y axes always measures vehicle acceleration/deceleration (throttle and brake response) and the Z axis always measures the cornering accelerations, no matter what the orientation of the device.

The result of this overall architecture is shown in Figure 6. This is a view of the entire data source tree, from the video camera to the internal data sources and the external vehicle ECU parameters, provided in one place.

Data And Video View Server

To view the recorded data on the data server, the data first needs to find its way onto the server. To get the recorded session data from the source (the smartphone) to the web-based server takes several steps. Video rendering and data fusion take up a lot of time and memory, so the data first needs to be uploaded from the smartphone to a space where it can be worked on by systems with far more resources than on a smartphone.

The repeatability of such a system is invaluable to the enthusiast and up-and-coming race driver alike, as individual runs can be viewed by any number of users and the data at each point of interest can be immediately cross-referenced to the time-synchronised video display. Figure 7 is an image of the typical output of a recorded and uploaded session.

The data from the recorded session is also displayed on a time-based oscilloscope-type display (see Figure 8 for an

example), which can be used to provide an overall view of the entire run. In Figure 8, the engine speed, vehicle speed and turbo boost pressure are all displayed. This type of view makes it possible to zoom in on various times for a more detailed analysis of the areas of interest and, with the video view, the exact location and circumstances of any area that gave rise to a slow corner, or better-than-expected lap time can be instantly reviewed.

With the data to hand, the types and formats of data display are almost limitless, allowing overlays of information from different sessions and, also, a scrolling function that shows data scrolling across the screen in sync with the video data – all of which can be seen along the pit lane of any F1 Grand Prix. ●



Figure 7: Typical rendered video and data display from a recorded TrackSpy session

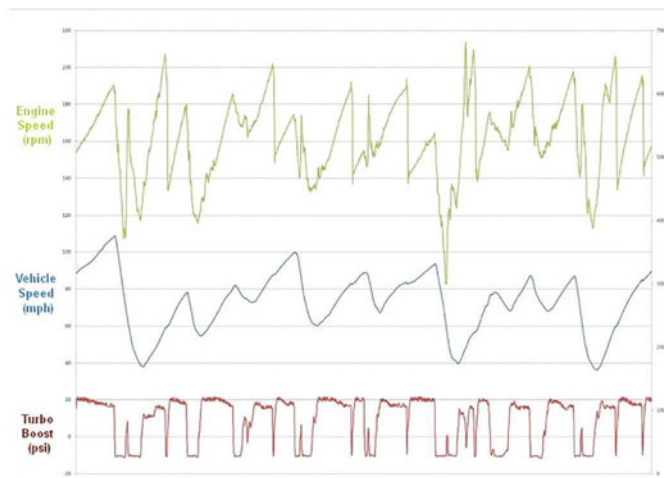


Figure 8: Time-based view of typical lap data

WT3000E – ONE OF THE MOST ACCURATE AND STABLE POWER ANALYSERS AROUND

Yokogawa has enhanced its industry-leading WT3000 precision power analyser to create the WT3000E: one of the most accurate and stable power analysers around.

The specified 45-65Hz accuracy for the WT3000E is 0.01% of reading plus 0.03% of range; figures based on RMS rather than waveform-peak values. The actual power measurement error due to an uncertainty of 0.03% of range in a WT3000E is less than 0.01% of that for a power meter based on peak values.

These accuracy levels are important for testing devices such as solar inverters, which are working at overall efficiencies of 90-96%. Also, the WT3000E provides the flexibility to mix 30A and 2A input current elements, enabling users to test the compliance of their products.

<http://tmi.yokogawa.com>



3.3V PCIE CLOCK GENERATORS FROM IDT NOW SHIPPING FROM MOUSER

Mouser Electronics is now shipping the 9FGL PCI Express (PCIe) clock generators from Integrated Device Technology (IDT). The 9FGL family of clock generators are members of IDT's 3.3V low-power PCIe series and support both common clock (CC) with or without spread spectrum and separate reference no-spread (SRnS) PCIe clocking architectures. Operating at 120mW and 130mW power consumption respectively, the 9FGL06 and 9FGL08 generators eliminate thermal concerns in high-performance consumer devices.

The IDT 9FGL PCIe clock generators generate low-power HCSL differential clock outputs in either 6-output (9FGL06) or 8-output (9FGL08) forms. All 9FGL clock generators feature support for two different spread spectrum levels plus an off function (0% spread). Direct connections to transmission lines and small VQFPN packages reduce board space and BOM.

www.mouser.com



POSITRONIC'S WATERPROOF COMBO-D CONNECTORS NOW AT ASTUTE ELECTRONICS

Astute Electronics is now stocking a waterproof version of Positronic's Combo-D mixed power/signal connector family. The WCBD series connectors, which have only just been launched, maintain a rating of IP67, making them ideally suited to non-sheltered and outdoor applications.

Combo-D connectors allow the mixing of power and signal in a single connector body capable of up to 40A per size 8 contact. The new WCBD styles feature a flexible, thermal-resistant adhesive seal and mounting accessories and hardware that prevent water and dust ingress into the enclosure. The materials used have been chosen to offer long-term corrosion resistance.

WCBD series connectors are available with vertical and right angle solder PCB terminations in two tail diameters and, optionally, with flying leads.

www.astute.co.uk



MOISTURE BLOCKING CONNECTOR NOW AVAILABLE FROM HYLEC-APL

Hylec-APL, the specialist supplier of electrical and electronic components, enclosures and rack systems, has announced the availability of the Techno TH392 waterproof connector with X-Dry technology. All cables are hydroscopic to some degree and will therefore absorb moisture that can collect inside the cable sheath, moving along its length and even crossing terminations in the cable. The X-Dry technology blocks moisture transmission across terminals, protecting electrical and electronic equipment.

The connectors have screw or piercing terminals and can accept cables with diameters from 8-12mm; maximum cable cross section is 4mm² for screw terminals and 0.5-1.5mm² for the piercing clip terminals. Water and dust protection is to IP68 at 3-bar pressure from 30 minutes to two hours. Rated current is 17.5A, 450V, and operating temperature from -40 to +125degC.

www.hylec-apl.com

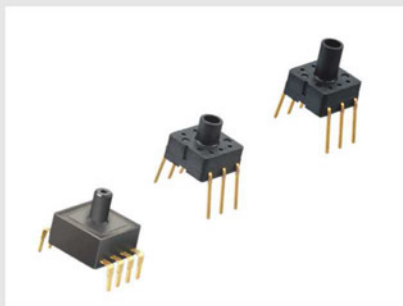


SEMICONDUCTOR SENSORS FROM PANASONIC HANDLE LOW-TO-VERY-HIGH PRESSURES

Panasonic Automotive and Industrial Systems has introduced a broad range of high-precision, miniature, semiconductor pressure sensors that can handle pressure. They are available with built-in amplification and temperature compensation circuitry (PS-A units). Precision sensors without amplifier and open wheatstone measurement bridge are also available and in ultra-miniaturised sizes (PS). In addition, there are versions with chamfered pins for improved ease of DIP pin insertion into PCBs.

The PS-A units come in three types. The standard type with a glass base can handle pressures from ±100kPa to -1000kPa with a total accuracy of ±1.25%. The economy type without a glass base is for 40kPa pressures and ±4% accuracy and the low-pressure version is for 6kPa and ±2.5% accuracy.

<http://eu.industrial.panasonic.com/>



COBHAM WIRELESS TM500 VALIDATES 600 MBPS LTE-A

Cobham Wireless, provider of mobile communication systems, announced that its industry-standard TM500 network test system now includes support for 256 QAM modulation – one of the key features of LTE-Advanced (LTE-A, also known as 4G), part of the 3GPP Release 12. The use of this modulation scheme in combination with carrier aggregation (CA) in the downlink boosts data rates up to 600Mbps and allows network operators to improve spectrum efficiency for mobile terminal devices under favourable operating conditions.

Increasing the modulation order to 256 QAM provides a 33% increase in LTE-A data rate compared with 64 QAM, and can be implemented when channel conditions are favourable and cells are sparsely populated.

Downlink 256 QAM support is available as a software upgrade for existing TM500 customers.

www.cobham.com



RS COMPONENTS INCREASES STOCK OF TOP-SELLING ALPHA WIRE PRODUCTS

RS Components (RS) has enhanced its stockholding of most-wanted product lines from Alpha Wire as part of a strategic initiative to ensure customers' favourite products are readily available for immediate dispatch and next-day delivery.

In total, RS supports over 3,000 product lines from leading global cable manufacturer Alpha Wire. Among Alpha Wire distributors, RS holds the widest product range and has now established an intelligent pro-active program to maximise support for all the best-selling Alpha Wire lines and ensure the most competitive prices.

The Alpha Wire portfolio consists of diverse general-purpose, data, power and networking cables, hook-up wire, and accessories such as heatshrink tubing, cable braid and copper tape. Product families including EcoGen, Xtra-Guard Industrial Series and Alpha Essentials provide a rich cable selection.

www.rs-online.com



LINEAR 3V HALL-EFFECT POSITION SENSOR IC WITH I2C OUTPUT

The new A1454 from Allegro MicroSystems Europe is a linear 3V Hall-effect sensor IC that provides a 12-bit digital I2C output that is proportional to the strength of the magnetic field that is present.

The A1454 is targeted at the consumer, industrial and instrumentation markets with end applications to include non-life-critical medical and consumer devices that require linear position sensing. The quiescent output value of the A1454 is set at mid-scale, and the device is available in two different factory-programmed sensitivity ranges: 2 LSB/G and 4 LSB/G. The sensitivity temperature coefficient is also factory programmed to support either neodymium or ferrite magnets.

The I2C address can be set by external resistors or programmed via EEPROM to support up to 127 unique I2C addresses, allowing for multiple ICs on the same bus.

www.allegromicro.com



MICROTHERMAL MASS FLOW METERS FOR G4 AND G6 SMART GAS METERS

Sensirion presents two new versions of the SGM70xx mass flow meter modules for G4 and G6 gas meters. These microthermal gas meter modules enable real-time gas flow monitoring and come with the additional advantages of high-level reliability and long-term stability. Onboard software in the mass flow meters guarantees precise gas quality compensation. As a result, the Swiss company Sensirion has further consolidated its expertise in the smart energy sector.

Until now, microthermal gas flow modules from Sensirion were available for G1.6 and G2.5 gas meters. Like the other versions of the SGM70xx series, the latest modules for the G4 and G6 gas meters stand out with high-level reliability, long-term stability and resistance to dust and dirt. Apart from this, they feature a standard connection and an I2C interface.

www.sensirion.com

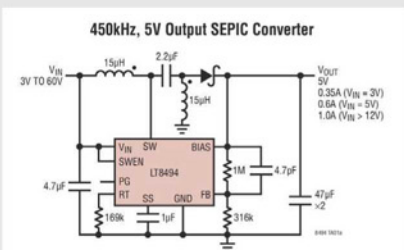


LINEAR'S 2A, 70V SEPIC/BOOST DC/DC CONVERTER OFFERS 7µA QUIESCENT CURRENT

Linear Technology launched the LT8494, a current mode, fixed frequency SEPIC/boost DC/DC converter with an internal 2A, 70V switch. Ultralow quiescent current of only 7µA makes the device ideal for always-on automotive or other industrial battery-powered systems.

The LT8494 starts up from an input voltage range of 2.5V to 32V and, once running, it operates from inputs from 1V to 60V, making it suitable for applications with input sources ranging from a single-cell Li-ion to automotive inputs. The LT8494 can be configured as a boost, SEPIC or flyback converter. Its switching frequency can be programmed via a single resistor between 250kHz and 1.5MHz, enabling designers to minimize external component sizes. The combination of a thermally-enhanced TSSOP-20E or 4mm x 4mm QFN package ensures a very compact footprint whilst minimizing solution cost.

www.linear.com



Telephone: 01840-770028
Fax: 01840-770705
7 Gavercoombe Park Tintagel, Cornwall PL34 0DS
www.kestrel-electronics.co.uk

KESTREL
Electronic Components Limited

100+ PRICES
many other PICS available

PIC12F508-I/SN	0.22	PIC18F46K22-I/PT	1.27
PIC12F510-I/SN	0.24	PIC18F65K22-I/PT	1.5
PIC12F615-I/SN	0.29	PIC18F67K22-I/PT	1.81
PIC16F616-I/SL	0.39	PIC18F87K22-I/PT	1.99
PIC12F629-I/SN	0.36	PIC18F4520-I/PT	2.05
PIC12F675-I/SN	0.37	PIC18F4525-I/PT	2.31
PIC16F628A-I/SO	0.69	PIC18F4620-I/PT	2.55
PIC16F818-I/SO	0.87	PIC18F4685-I/PT	3.49
PIC16F819-I/SO	0.95	PIC18F6621-I/PT	3.95
PIC16F873A-I/SO	1.55	PIC18F6622-I/PT	3.48
PIC16F876A-I/SO	2.09	PIC18F6627-I/PT	3.53
PIC16F877A-I/PT	1.99	PIC18F6722-I/PT	3.81
PIC16F882-I/SO	0.73	PIC18F8722-I/PT	3.75
PIC16F883-I/SO	0.65	PIC18F8723-I/PT	3.72
PIC16F883-I/SS	0.71	PIC16F508-I/P	0.3
PIC16F883-I/SP	0.95	PIC12F629-I/P	0.45
PIC16F886-I/SS	0.88	PIC12F675-I/P	0.53
PIC16F886-I/SP	1.05	PIC16F84A-04/P	1.68
PIC16F887-I/PT	0.84	PIC16F628A-I/P	0.82
PIC16F1933-I/SS	0.59	PIC16F876A-I/SP	2.25
PIC18F45K20-I/PT	0.98	PIC16F877A-I/P	1.89

**We can also supply Maxim/Dallas, Lattice, Linear Tech
PLEASE VISIT OUR WEB SITE FOR FULL LIST**

Ever wished for a better bench scope?

FROST & SULLIVAN

2015

BEST
PRACTICES
AWARD

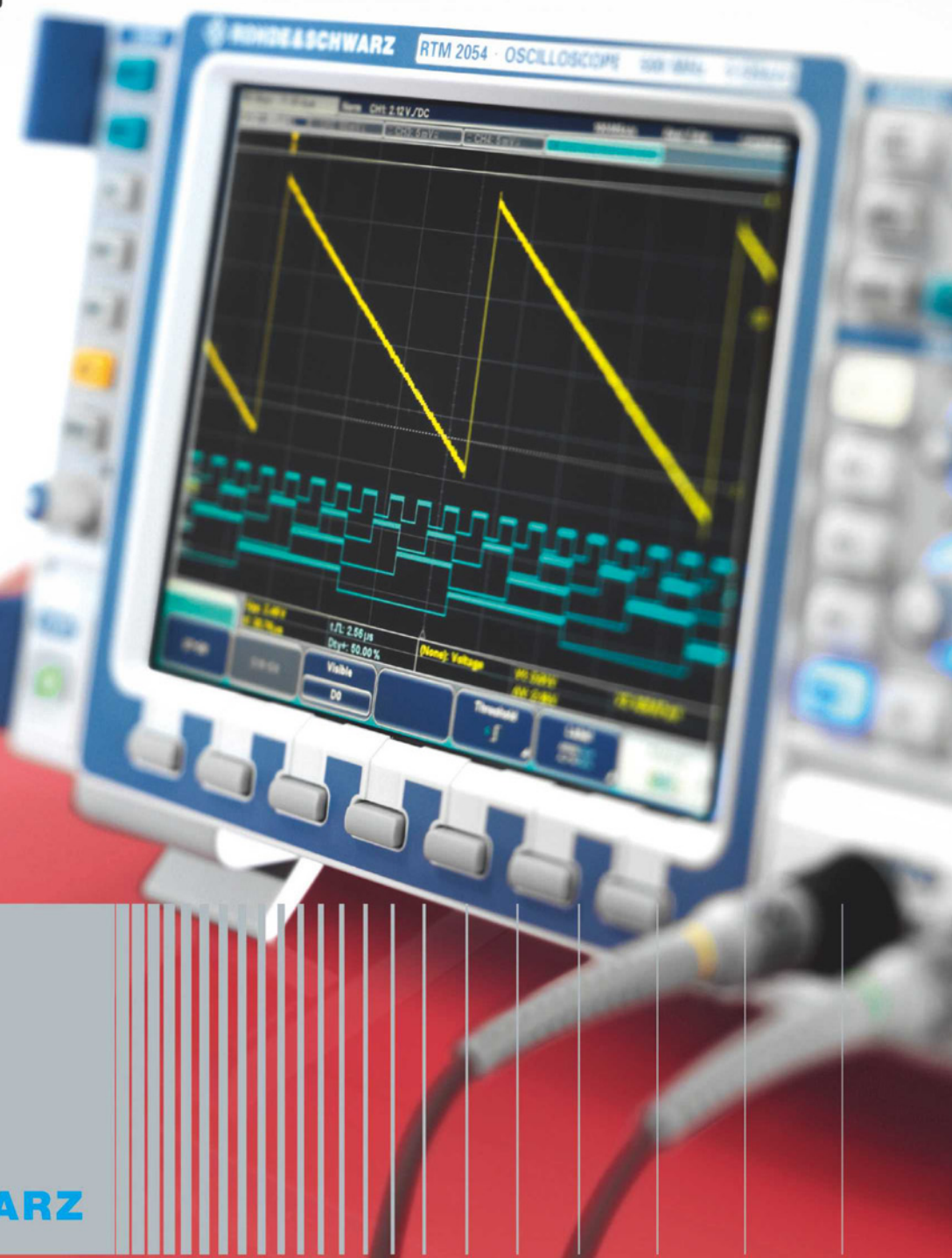
Global Oscilloscopes
Competitive Strategy
Innovation and
Leadership Award

The new R&S®RTM2000: Turn on. Measure.

Ease of use, combined with fast and reliable results, is precisely what users get with the R&S®RTM bench oscilloscope. While other oscilloscopes are still booting up, the R&S®RTM is already displaying signals that would otherwise be lost in the noise and evaluating results. All on one screen with two displays, with lightning fast functions.

For more information, visit:

www.scope-of-the-art.com/ad/rtm2000



ROHDE & SCHWARZ