Embedded Hardware Design For An Autonomous Electric Vehicle

Henry Jenkins
hvj10@uclive.ac.nz

Coauthors: Simon Richards, Zachary Taylor, Wim Looman and Dr Andrew Bainbridge-Smith
scr52@uclive.ac.nz zjt14@uclive.ac.nz wgl18@uclive.ac.nz
andrew.bainbridge-smith@canterbury.ac.nz

Department of Electrical and Computer Engineering
University of Canterbury
Christchurch, New Zealand

Abstract—This report details the design and production of a hardware platform to for an autonimous go-kart. The system includes seven printed circuit boards (PCBs). Five of the PCBs communicate via a CAN bus to collect data and control other circuitry, while the other two PCBs control actuators for steering and brake control. The system is based around Atmel AT91SAM7XC micro controllers along with various other electronics. The PCBs built for this system are four layers so were manufactured in America by Advanced Circuits, then populated and tested at the University of Canterbury.

I. INTRODUCTION

Currently the department of Electrical and Computer Engineering at the University of Canterbury has several go-karts. These go-karts were purchased as petrol go-karts then retrofitted with control electronics, power electronics, lead-acid batteries and a DC electric motor. The goal of this project was to take one of these existing electric go-karts and have it drive it's self autonomously.

An autonomous vehicle is a driver-less vehicle that is able to navigate from one location to another set location by controlling actuators and motors. This is achieved by gathering data from the surrounding environment in order to find and track the vehicle's current location. While the vehicle can have communications off the vehicle, like off loading signal processing, it is ideal to have the vehicle completely self contained. This, however, does not dismiss using incoming radio signals. For example using Global Positioning System (GPS) would be perfectly acceptable.

The control systems, both mechanical and electrical, must interface with existing hardware on the go-kart. The project aimed to select appropriate actuators, motion and distance sensors, develop a navigation system, interface to the existing control systems and interface to a central computing platform. As the kart also was required for other projects, it was also a requirement to able return the kart to it's initial electric go-kart state within an hour.

To be able to drive a go-kart autonomously, first the kart must be able drive according to given digital instructions. So the initial milestone is to build embedded hardware systems that is able to control the go-kart's main motor and drive actuators for both the brake and steering. Because the time line of this project spans approximately only 24 weeks, a sub-goal was set of having *drive-by-wire* go-kart by the end of the project.

To allow high performance, high level signal processing the control system must be able interface to a computer based platform. This platform will be able to run processor intensive tasks, like image processing for computer vision [1], or a Bayes filter for estimating the kart's current location before it receives accurate sensor data.

II. BACKGROUND

To interface to the go-kart, actuators were required to be placed at both the front of the kart, as well as interfacing to the power electronics in the rear.

A. The Go-kart

The department has 5 electric go-karts that were purchased for the specific purpose of a assignment and project platform, one of these shown in Fig. 1. Throughout the year several, mostly power electronics, assignments are based on the Junoir Sport JS80IIR go-kart [2]. These go-karts are 1.7 meters in length, can hold two passengers and came with a 80cc petrol motor. The petrol motor has since been replaced with a DC electric motor and batteries.

B. Power Supply

The go-kart has a single power source for the whole vehicle. Two lead-acid batteries provide power for the existing DC drive motor and power electronics. These batteries can handle large currents at a voltage ranging up to 26 volts. The batteries have also been made accessible to be used for the autonomous vehicle control system.

Fig. 1. Electrical and Computer Engineering go-kart

C. Power Electronics

To control the DC Drive motor, the department has designed and built three PCBs to switch and regulate currents going through the motor. Figure 2 shows the three PCBs, where from left to right they are known as the *student board*, *control safety board* and *power electronics board*. The *student board* is a 5V logic board that takes the position of the accelerator pedal, current sensor data and outputs a PWM signal for the *control safety board*. The control safety board is there to make sure the *power electronics board* is not incorrectly driven, which would damage the go-kart.
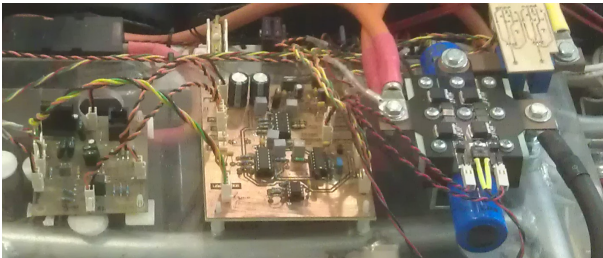

Fig. 2. Power and control electronics for go-karts DC motor

D. Actuator Control

To control the steering position and apply the brakes two actuators were purchased for the project.

1) Steering: The steering motor is a 24V DC motor that has a step down gear box with an output ratio of 393:1. On the top end of the motor, there is a magnetic encoder [3] that outputs 18 pulses per motor revolution. This is that it outputs 7,074 pulses per 360 degrees of rotation on the steering wheel. At peak load, the steering motor can draw up to 10Amps continuously.

2) Brake: To control the position of the brake a linear actuator is used. The linear actuator has an extension of 101.6mm and can travel at 20mm/s. It provides absolute position feedback via a $10k\Omega$ potentiometer. As with the steering motor, the brake linear actuator ran at 24V but only draws a peak current of 1.3Amps.

E. Communication With A Laptop

Because most laptops have dropped all serial RS-232 ports, the only reliable option for serial communications is to use USB.

F. Schematic & PCB design tools

To generate the PCBs and schematics, Altium Designer was used [4]. Altium Designer is a complex electronics design package that can be used from a variety of tasks. However in this project only the schematic layout and PCB design tools were used.

III. DESIGN

To avoid problems of running long signal wires, especially near the large DC go-kart motor, it was decided to distribute the electronic control hardware. This also has the benefit that the system is extensible including being able add new sensors later on with plenty of spare IO spread around the kart.

To make the design simpler and cheaper the same design was used for all the main PCBs. Just not populating certain components where they are un-nessary. For example, the USB connector was only placed on the *comms board* as it is the only board that requires to talk to a computer.

A. Board locations

To prevent long runs of digital logic in the go-kart there were five PCBs placed around the go-kart. The locations of the boards are:

- Steering - Placed where the steering wheel is, next to the steering actuator.
- Brake - Directly next to the brake actor where the brake pedal normally is.
- Comms - On the go-kart seat with the laptop next to it.
- Motor - Connected to the power electronics controlling the main DC motor
- Sensor - Near the rear axle collecting speed data from the inductance sensor.

Figure 3 shows a sketch of where the PCBs are located.

B. Communications Network

Because the go-kart was required to be controlled by distributed hardware, some way of having the hardware communicate was required. To solve this problem, Local Interconnect Network (LIN Bus), Controller Area Network (CAN Bus), Universal Asynchronous Receiver/Transmitter (UART) and Serial Peripheral Interface Bus (SPI) were evaluated as viable options. Both LIN and CAN are serial communication buses designed for use in vehicle and automation systems, where UART and SPI are designed as simple serial communications.
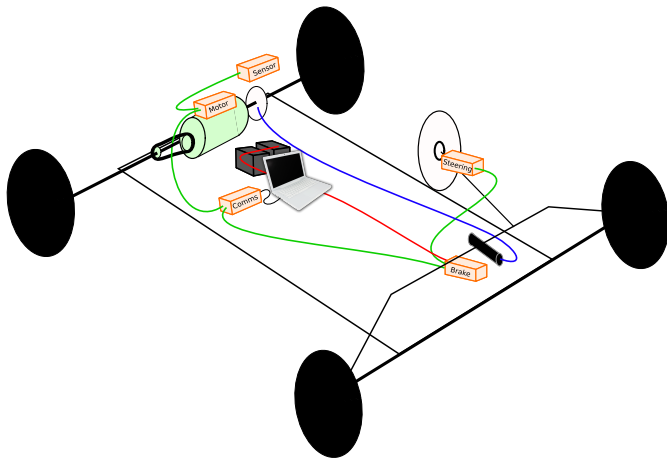
Fig. 3.   Locations of PCBs for go-kart

LIN LIN is a single wire communications standard which supports speeds of up to 19.2kbit/s [5]. It is a master slave design which supports up to sixteen slaves. The data frame on the LIN Bus is variable length of either 2, 4 or 8 bytes. There is no collision detection, however data checksumming and error detection is implemented in the standard. LIN was designed in 1999 with further LIN standards released in 2002 and 2003.

CAN CAN 2.0 is a master-master broadcast serial bus standard. It supports half-duplex data rates of up to 1Mbit/s. The CAN Bus uses a differential pair with the traffic encoded in non return to zero. Each packet is formatted with a header, including receivers address, followed by zero to eight bytes of data. Robert Bosch GmbH [6] developed the standard in 1983, but now days the ISO11989 standard also describes CAN 2.0 [7], [8].

UART Using UART was dropped as an idea rather early on. This is because it does not support the error correction that was needed for the project. Another area it was lacking is being expandable. A new UART would be required for each new node or device.

SPI For much the same reasons UART was not used for our central communications bus, SPI was also not appropriate.

It was decided that using a CAN Bus would be the best option. Some major factors that influenced this choice over LIN were; the CAN standard had better documentation, more nodes can be added to the network later on with little extra effort, hardware level collision and error detection and CAN controller supported by Atmel SAM7 series micro controller. Atmel also provided a CAN transceiver IC [9], however this could have been replaced by a IC made by Texas Instruments [10].

Although both UART and SPI were not used for the system bus, both can be used for local sensors, or other expansion modules.

C. Micro Controller

The main requirements of the micro controller are that it supports both a CAN controller and hardware USB. Also because previous projects had used either the Atmel ATMega or the Atmel AT91SAM7 micro controllers, it was of preference to use these rather than learning a new architecture and data sheet format.

D. Motor Drivers

To drive the actuators for the steering and brake we decided to use a motor driver IC. By basing our motor driver circuit specification on the current draw of the steering motor, then reused the same design for the brake linear actuator we could use the same circuit for both actuators. This was done both to save time in the hardware design, but also time in software later on [11]. The motor driver circuits were placed on their own small PCB to save space on the main PCBs and also provide more opportunity for heat dissipation if this were to become a problem.

The motor driver that was used is the DRV8432 made by Texas Instruments [12]. It is a dual full bridge PWM motor driver. It has a split logic and driving voltage, so it is driven directly from the batteries, but the control logic is at 3.3V. It has a peak voltage of 50V and a maximum current of 12A which is more than we need for either of the actuators. Figure 4 shows one of the two motor driver boards with a heat sink covering the motor driver IC.
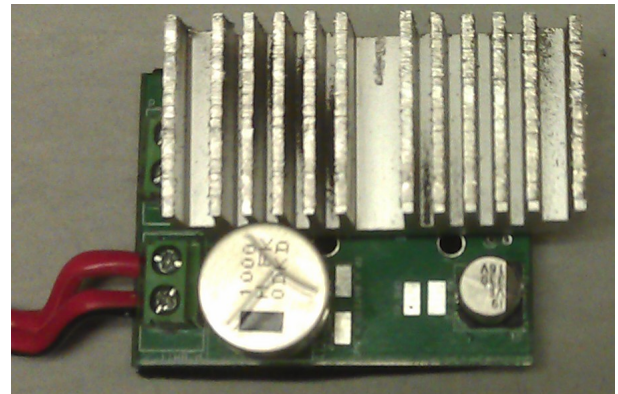

Fig. 4.   Motor driver board for driving actuators

E. Expansion IO

To allow the hardware to be expandable, two 3.3V UARTs, one 3.3V SPI and one 5V logic SPI are exposed on each PCB. Each serial port has it's own PicoBlade connector [13] as well as all the power supplies are exposed via another connector. All the serial port pins can also be set to be GPIO to allow them to be used for other purposes too.

The 5V logic SPI is enabled via the use of a logic level shifter chip the MAX3001E made by Maxim [14]. To complement the 5V spi, a 5V external ADC is also on the boards. This was placed on the boards for the purpose of replacing the student board, but can be used for other analogue input too.

F. Encoder Reading

Because getting both speed and directions from an encoder signal in software is quite tricky, a quadrature clock converter was placed on the main steering PCB. This takes in the encoder pulses and out puts a single pulse and which direction it is rotating in. Which allows a simple counter and incrementing or de-incrementing it to keep the actuators position.

G. Debugging

To make debugging easier, epically in the early stages of development, a two digit seven segment display and 4 push buttons are placed on each main PCB. The display is driven by a display driver. This display driver takes a four bit number in parallel and converts it into the segment array lines. It is multiplexed with the second segment so by switching between the displays faster than 20Hz allows both digits can be used.

H. On Board Power

On each board there is four different voltage five levels. There is the battery voltage, which is only used by the voltage regulators, 3.3V, 5V, 12V and ground. Because we used a 4 layer PCB design, excluding the battery voltage, each of the voltage levels were spread across a whole plane on the PCB allowing better coupling between them and ground.

1) 3.3 Volts: This was used by most of the logic components including the SAM7 micro controller. As less than 300mA was required and having a smooth, non-rippled, voltage would ensure the micro controller ran well, a linear voltage regulator was used.

2) 5 Volts: The main reason for having 5 volts on the boards was to allow a 3.3V to 5V logic translator to be used. This level shifter is required by the student board in the power electronics. A linear voltage regulator was used here also.

3) 12 Volts: Both the inductance sensor, which is used for speed detection, and the motor driver required 12 volts. Both of these applications use less than 150mA so a Texas Instruments TPS7A4901 linear voltage regulator was used [15].

IV. IMPLEMENTATION

The Main PCBs are four layer PCBs that were designed using Altium software. They are approximately 120mm long by 60mm wide.

A. Schematics

Using Altium Designer the schematics were laid out. Two techniques were used to minimise mistakes in the PCB design. One was to divide all the schematic sections onto separate sheets. The other was to be methodical and make notes on the sheets if something was not complete. Part of being methodical also meant making sure all components were well labelled, even some with comments next to them to help explain their existence. This meant when the schematics were printed off and reviewed by the whole team, mistakes could be easily spotted.

Figure 5 shows the top level schematic with the MCU in the middle of the page. The green boxes on the sheet show where other sub sheets have been included and wired up. For example, doing this all the decoupling capacitors for the micro controller are essentially hidden in one of these sub sheets.

B. PCB Design

The PCB layout design work was also done in Altium Designer. While the tool does not automatically do the layout for the user, it does provides many useful tools to aid the process. By setting up the design rules to match the manufactures specifications, the manufacture process is sped up eliminating them rejecting your design because of design rule violations.

The main technique that was used when laying out the PCB was to arrange clusters of components that are all related. Then place these modules on the PCB area. This also allowed the placing of decoupling capacitors in the correct locations next to the pin that required the decoupling.

One key rule that was stuck to when laying the PCB was to only put small components on the bottom. This allowed the population of the whole bottom side of the PCBs then when populating the top side, the components would not fall off the bottom in the oven. It also meant we could fit the PCBs in the lid of the enclosure that we designed the boards to fit in. The small resistors and capacitors don't even touch the inside of the lid.

C. Components

Where possible surface mount technology (SMT) components were selected. This has the benefit that it minimises the amount of drilling required to be done by the manufacturer. It is also easier to source SMT components. In the final design only some connectors are non-SMT.

D. PCB Manufacture

To allow four layer PCBs to be used, the PCBs needed to be manufactured off-shore. Because of recommendations by staff in the department, either Advanced Circuits in America [16] or OurPCB Tech in China [17] were our best options. Because of price, Advanced Circuits were used. All up for 3 panels, which included all 5 main PCBs and two motor driver daughter boards, the charge was 362 USD including shipping. The panels were manually cut into the individual PCBs, using a band saw.

E. PCB Population

Component population was done initially using a PCB oven, then reverting back to hand soldering once the bulk of the components were on the PCBs. Population was done in stages, to allow testing of the boards at each stage. Figure 6 shows the brake PCB with all it's components on it.

F. Hardware Testing

Before the PCBs even had any components put on them they were tested for manufacturing and design flaws. Simply using a multi meter, checking to make sure the power planes were not shorted and other key pads were connected as they should have been. Then continued to be tested at every stage especially making sure the power and ground planes were not
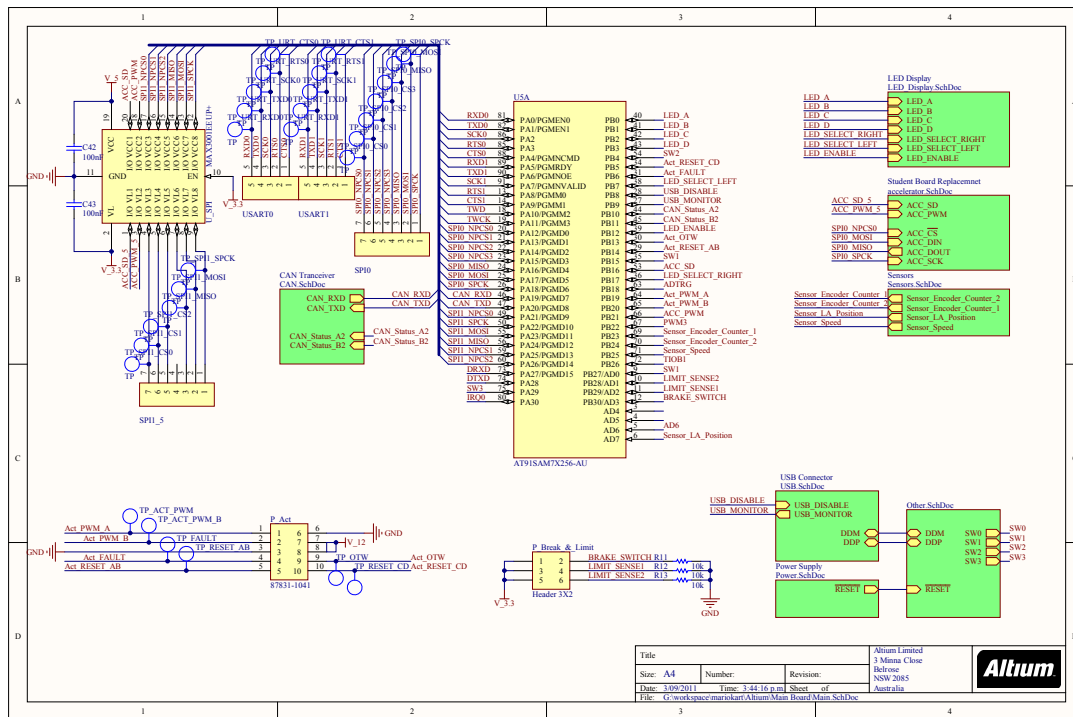
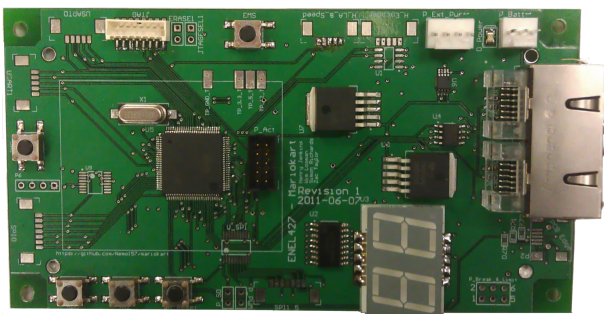Fig. 5. Top level schematic for the main PCB.

Fig. 6. The populated brake PCB

shorted was a key concept. At several stages shorted pins were picked up because of this testing.

Once the PCBs had their micro controller placed on its PCB, led test code was written to ensure we were able to program and control the board. This also helped to pick up shoddy soldering on a couple of the micro controllers. The same was repeted for the CAN Bus, push buttons and USB.

V. DISCUSSION

In general the design process produced good stable and usable boards. However there were a few areas that needed improving on.

A. Debug Port

When the schematic was being set out the debug unit from the SAM7 were left with no header leading to them. At

the time it was thought that they would be unnecessary and then were forgotten about. This meant all the debugging was required to be done via a UART port and the display. While this wasn't a bad solution to the problem, it would have been better to have access to the debug unit.

B. Test points

At the last minute in the PCB design about 100 test points were added to the bottom of the PCB. This was very useful and one of the better design decisions with the PCB. The test points gave access to many different signals on the board with test point probes able to be clipped onto them. The only thing that should be done differently with the test points is placing more ground test points around the board. This would allow more than one oscilloscope cable to be hooked up to the ground on the board.

C. Power Supply Selector

Making the decision to put battery voltage over the same cables that the CAN bus uses made the system work well. However when developing and debugging, it would have been ideal to be able to disconnect the power coming in from the CAN bus cable. This would mean that an individual board could be power cycled without having to power cycle all of them. This could have been done by simply putting two jumpers on the PCBs to connect specific ones to enable either the CAN bus power or the battery header power.

D. Connectors

To save space the Molex Pico Blade [13] connectors were used. However after using them this was a bad choice. They are very hard to make cables for, then brake a short time after. This is frustrating when it takes so long to make them, yet they don't last long. While using bigger more traditional connectors would have taken up more space on the PCBs, it would have saved a lot of time while developing with the boards.

VI. CONCLUSIONS

The design process used for the schematic and PCB layout worked well providing a set of very functional and reliable PCBs. They have been shown to all communicate together via the CAN Bus protocol and drive both the steering and brake actuators.

Since the project has been in the software development stage for some time now, there have only been three mistakes in the hardware found. All of these mistakes have had relatively easy fixes. The worst of the three is one side of the quadrature clock converter's pins were reversed. This was fixed by soldering the IC on upside down and then wiring the pins over to the correct pins. The second worst was a miss placed decoupling capacitor on the *avref* pin of the SAM7. This was simply replaced with a zero ohm resistor to solve the problem. The third problem was that the UART connectors don't have a ground pin. To get around this the *sclk* pin in tied low in software to act as a ground.

Using the same design for all of the five main PCBs worked well. Doing this saved time in the design, ordering and population. This also saved money by being able panellise the boards which would have cost 50 USD more if wasn't done. By not having to spend time on making five different designs, more time was able to be spent checking for mistakes in the design. If there were multiple designs more mistakes would have happened.

The CAN Bus produced clean signals with all nodes connected. Even with the CAN bus running at 1Mb/s its maximum speed. While the bandwidth of the CAN Bus is about 1000 times more than required for the current application of a drive-by-wire go-kart, it does mean the CAN bus can be expanded for future use with sensors and other devices generating traffic on the network. If noise were to become a problem, the bus is able to run at a slower rate potentially solving this issue.

REFERENCES

[1] G. Bradski, "The opencv library," *Dr Dobbs Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000. [Online]. Available: http://opencv.willowgarage.com

[2] *Junoir Sport JS80IIR Owners Manual*, Junoir Sport.

[3] National Instruments. (2011) Using quadrature encoders with e series daq boards. [Online]. Available: http://zone.ni.com/devzone/cda/tut/p/id/4623#toc1

[4] Altium Limited. (2011) Altium live. [Online]. Available: http://live.altium.com/#software

[5] LIN Administration. (2011) Lin specification. [Online]. Available: http://www.lin-subbus.org/index.php?pid=9\&lang=en\&sid=62a662e7da7a4e8efed2e436985bd028

[6] Robert Bosch GmbH. (2011) Bosch worldwide. [Online]. Available: http://www.bosch.com

[7] *CAN Specification, Version 2.0*, Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart, Sept 1991.

[8] OurPCB Tech Limited. (2011) Iso 11898-1:2003 - road vehicles – controller area network (can) – part 1: Data link layer and physical signalling. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=33422

[9] *ATA6660, High-speed Can Transceiver*, 4582nd ed., Atmel Corporation, 2325 Orchard Parkway San Jose, CA 95131 USA, February 2008.

[10] *SN65HVD232 3.3-V CAN Transceivers*, K ed., Texas Instruments, Dallas, Texas, USA, February 2011.

[11] W. G. Looman, "Software engineering practices in the mariokart system," 2011. [Online]. Available: https://raw.github.com/team-ramrod/mariokart/master/Documentation/ScientificReport/Wim/report.pdf

[12] *DRV8412 Dual Full Bridge PWM Motor Driver*, C ed., Texas Instruments, Dallas, Texas, USA, May 2010.

[13] Molex Limited. (2011) Picoblade 1.25mm (.049) pitch connector system - molex. [Online]. Available: http://www.molex.com/molex/products/family?key=picoblade&channel=products&chanName=family&pageTitle=Introduction

[14] *MAX3001E +1.2V to +5.5V, 15kV ESD-Protected, 0.1A, 35Mbps, 8-Channel Level Translators*, 4th ed., Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA, USA, December 2006.

[15] *TPS7A4901 +36V, +150mA, Ultralow-Noise, Positive Linear Regulator*, B ed., Texas Instruments, Dallas, Texas, USA, Jan 2010.

[16] Advanced Circuits Limited. (2011) Pcb manufacture and assembly. [Online]. Available: http://4pcb.com/

[17] OurPCB Tech Limited. (2011) Pcb manufacturing — pcb assembling & components procurement service. [Online]. Available: http://ourpcb.com/