|  |
| --- |
| Drive by noise equipment |
| MIRA TTC040 equipment |
| D. J. O'Boy |

Contents

[1 Introduction 1](#_Toc493245942)

[2 Sound pressure meter 1](#_Toc493245943)

[2.1 Sound pressure meter connector to trackside transmitter cable 1](#_Toc493245944)

[2.2 Sound pressure meter power connector cable 1](#_Toc493245945)

[3 Light gates (gate 1 and gate 2) 1](#_Toc493245946)

[3.1 Cable from track transmitter to gate 1 (duplicate for gate 2) 2](#_Toc493245947)

[4 Telemetry radio options 2](#_Toc493245948)

[4.1 Equipment connection diagram using plan A 3](#_Toc493245949)

[4.1.1 Telemetry radios using plan A: Track setup 5](#_Toc493245950)

[4.1.2 Setup network for transmitter radio 8](#_Toc493245951)

[4.1.3 Battery booster power pack to on track transmitter cable 8](#_Toc493245952)

[4.1.4 Telemetry radios using plan A: In car setup 8](#_Toc493245953)

[4.1.5 Setup network for receiver radio 10](#_Toc493245954)

[4.2 Equipment connection diagram using plan B 11](#_Toc493245955)

[5 Range against errors generated 11](#_Toc493245956)

[6 To be completed 11](#_Toc493245957)

[7 amp hour batteries + wireless router + antenna 12](#_Toc493245958)

# Introduction

This document contains a description of the telemetry equipment and specifications used in the undergraduate and postgraduate drive-by-noise assessment at MIRA proving ground.

MIRA proving ground has a short noise track where ISO362 measurements can be carried out. A sound pressure meter (measuring A-weighted sound pressure levels in dB[A]) is positioned to the side of the track.

Requirements: Record the sound pressure level of the vehicle as it passes, the entry and exit time of the vehicle in the test area and send this information in real time to a data logger onboard the vehicle.

The sound pressure meters are Larson Davis LXT modules with a 2.5mm output jack (AC and DC is available as an output.

Connectors should typically be LEMO latch type L.

# Sound pressure meter

The Larson Davis sound pressure meter is set to A-weighted, RMS averaged with linear, fast integration. Whilst the students start and stop the devices according to when they believe the vehicle is in the zone and thus saves the peak level, there is an output jack which constantly outputs the sound pressure.

|  |  |
| --- | --- |
| Parameter | Value |
| DC output voltage range [V] | 0-3 (0-300dB)  Due to the dB scale, V0 approximately SPL/100  Sensitivity = 0.01V/dB with resolution of 0.001V. |
| DC output impedance | 3650 ohms |
| Connector | 2.5mm stereo jack  Tip: DC out  Ring: AC out  Sleeve: Ground |

Table : Output socket on the sound pressure meter

## Sound pressure meter connector to trackside transmitter cable

2.5mm stereo Jack (male) using DC and ground connections to male LEMO latch connector (2 wires).

## Sound pressure meter power connector cable

Male Mini USB to standard male USB (plugs into the sound pressure meter and USB socket on the car battery booster pack).

# Light gates (gate 1 and gate 2)

Light gate specifications to be determined. Requires a width of beam to be broken of approximately 5metres. Output voltage between 0-5V for use in the Arduino ADC.

Power to be provided through the connecting cable (Arduino transmitter to gates) which comes from the battery booster pack.

The light beam is transmitted and received by the same unit, using a reflector on one side of the track. A typical solution would be as shown:

<http://uk.rs-online.com/web/p/photoelectric-sensors/5124640/>

http://uk.farnell.com/schneider-electric-telemecanique/xuk1apanl2/photoelectric-sensor-7m-pnp/dp/2765982

It would need to operate at lower voltages however.

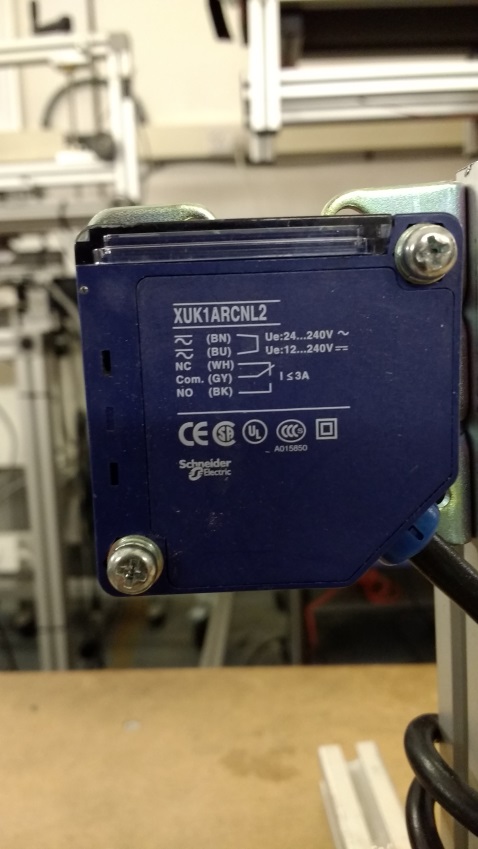


Figure : Proposed on track gate hardware

## Cable from track transmitter to gate 1 (duplicate for gate 2)

Male LEMO latch connector on transmitter box (3 wire), +12V or +9V, GND, V0, to Male LEMO latch connector on Gate box.

Estimated length of cable ~ 16m.

Power consumption ~ 200mAmps.

# Telemetry radio options

* Plan A: Trackside radio to in car receiver (no other substantial equipment)
* Plan B: Trackside radio to other trackside radio receiver. This latter is connected to a National Instruments Ethernet voltage acquisition system and communication to the car is via 2.4GHz wireless router.

## Equipment connection diagram using plan A

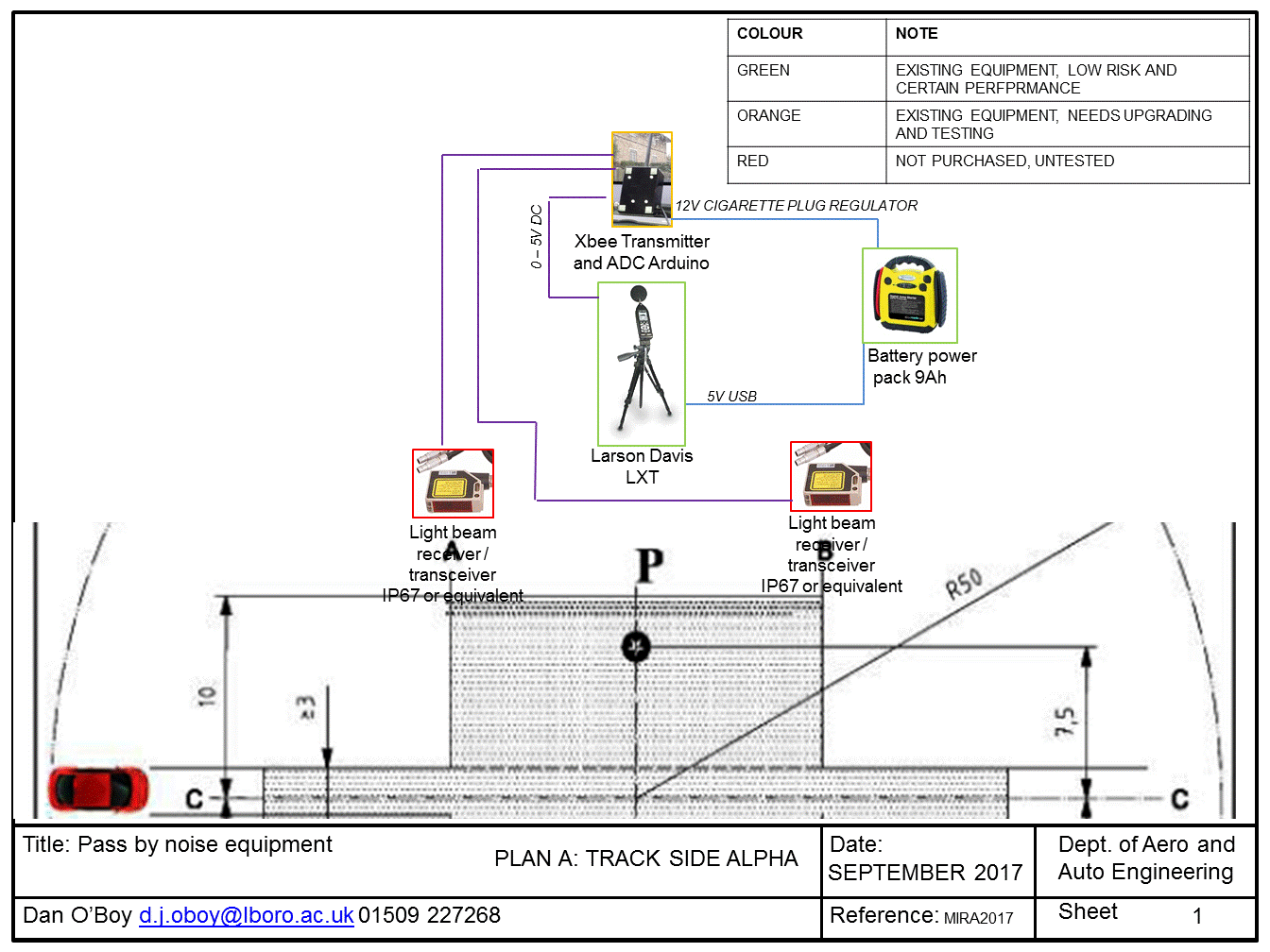


Figure : Plan A - On track hardware

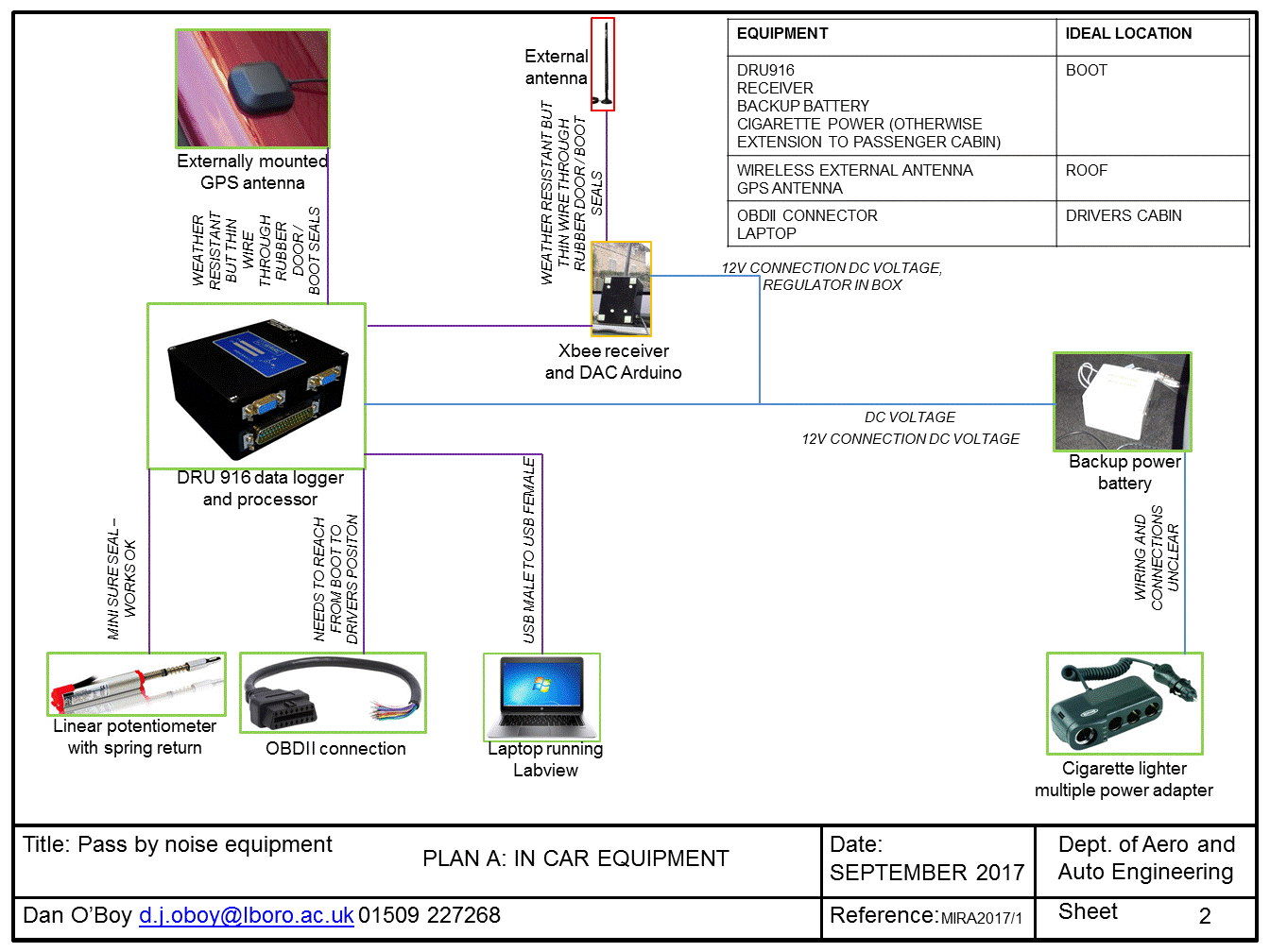


Figure : Plan A – In car hardware

### Telemetry radios using plan A: Track setup

To communicate between the track and the car (sound pressure meter readings and the track entry and exit point status (called gates 1 and 2 respectively), two telemetry radios are employed, using 2.4GHz serial data XBee radios.

The overall box devices are an Arduino Uno board with XBee wireless proto shield and XBee S2C radio. In order to determine the sound pressure level, an additional ADC board is used [ADS1115].

The exterior of the box has the following:

|  |  |
| --- | --- |
| Equipment | Specification |
| 12V DC input. | Female Lemo latch connector (2 wire). |
| Power switch. | On / off. |
| LED1 | Simple led |
| LED2 | Simple led |
| Gate 1 connections | Female Lemo (4 wire) |
| Gate 2 connections | Female Lemo (4 wire) |
| Sound pressure meter | Female Lemo (2 wire) |

Table : Exterior connections on the track transmitter box

The schematic of the circuit inside the box is as follows.

* All connections to the Arduino board should be through headers to enable the circuit to be replaced.
* USB socket needs to have room to insert a cable for reprogramming.
* Existing installation is shown below to illustrate difficulty in replacing components.

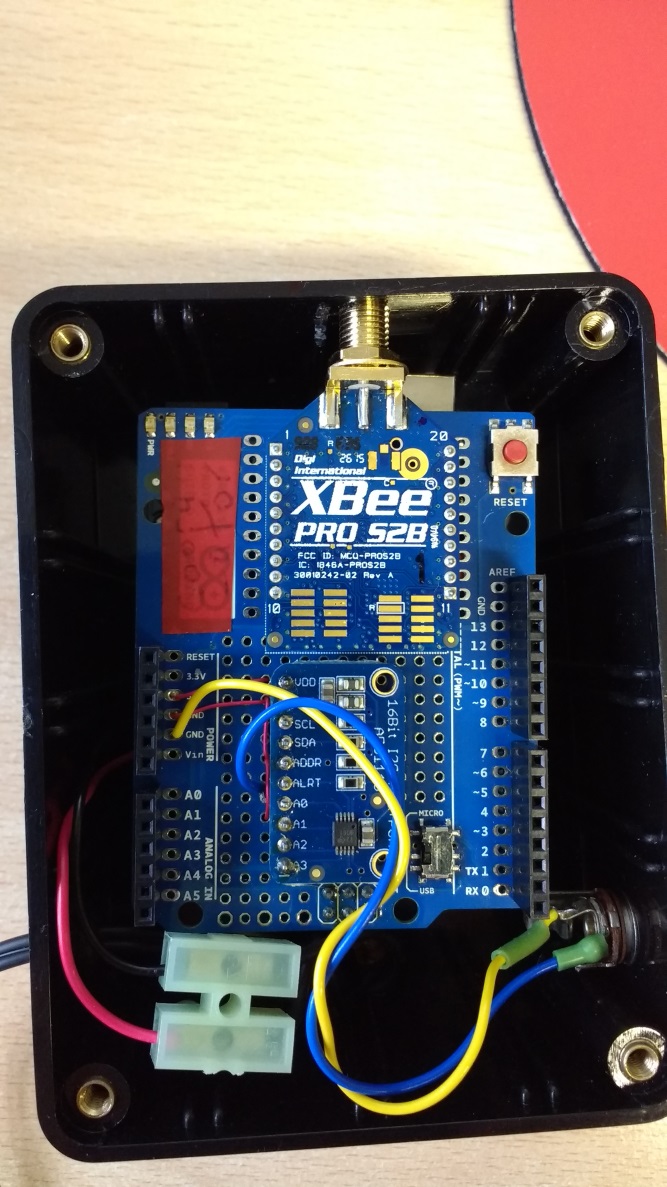


Figure : Plan A – Existing on track transmission box assembly

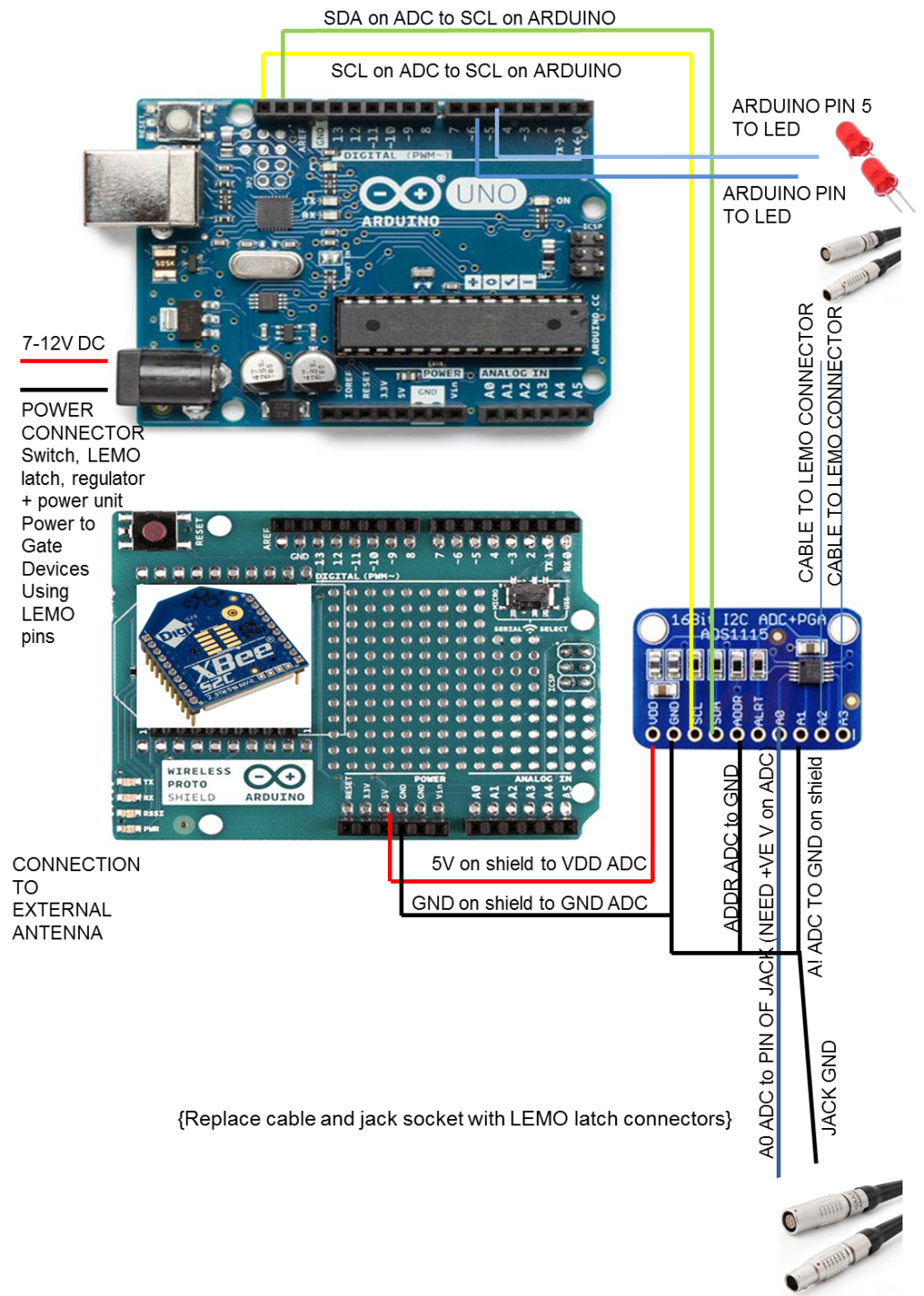


Figure : On track transmitter box internals

Sound pressure meter output assumed to be 0-5V DC. Input ADS1115 voltage on A0 assumed to be 0-5VDC. By connecting ADDR pin on the ADC to ground, it uses the address 0x48 (1001000). We use the option of measuring a single ended voltage between A0 and GND.

Although it is a 16 bit device, one of these bits is used to represent the sign (single ended are always positive). Hence the actual range is only 15 bit. So the range of numbers representing the voltages goes from 0 to 32767 (32768 values).

The ADC has a programmable gain amplifier (PGA) which sets the range of the voltages that it can measure (however, you still can’t go past VDD from the Arduino without damaging it so it isn’t a full measurement range).

With GAIN\_TWOTHIRDS the value 32767 corresponds to 6.144V. Hence 1 bit =0.1875mV. After testing, it could be that the range could be reduced to GAIN\_TWO.

|  |  |
| --- | --- |
| PROGRAMMABLE GAIN AMPLIFIER SETTING | VOLTAGE MEASUREMENT RANGE [V] |
| GAIN\_TWOTHIRDS | (for an input range of +/- 6.144V) |
| GAIN\_ONE | (for an input range of +/-4.096V) |
| GAIN\_TWO | (for an input range of +/-2.048V) |
| GAIN\_FOUR | (for an input range of +/-1.024V) |
| GAIN\_EIGHT | (for an input range of +/-0.512V) |
| GAIN\_SIXTEEN | (for an input range of +/-0.256V) |

Table : Analogue to digital converter gain settings

Once the unsigned integer has been obtained, it needs to be transmitted through the xbee to the receiver. It is converted into two bytes (high and low), see documentation in <https://www.safaribooksonline.com/library/view/arduino-cookbook/9781449399368/ch04.html>

Use the Serial.write(lowbyte(SPMAmplitude)) command.

We are looking for a data stream of around 10Hz = 0.1 seconds per sample. So we need to send two / four bytes in that time. A delay of 20ms can be added to allow the xbee to transmit. Assuming an ADC conversion time of 10ms still leaves room for the specification. (It also means it may be possible to include the start / end triggers for the light gates).

Two bytes are used for the integer, but there also needs to be synchronisation (or some method to determine when the start and end of the number is (i.e. whether the bytes are correct). Serial.write sends binary data to the serial port. The two bytes for the 16bit integer go from 0 to 32767. The latter is split into high and low

32767 = 127\*256 + 255, i.e. high byte is 127 and low byte is 255.

The high byte will never be greater than 127, so getting two values above this in succession can show the end of a message. We send two identical values of between 128-131 to denote the end of a message, where the difference relates to the gate condition (thus using the synchronisation bytes to also transmit information on the track condition).

|  |  |  |
| --- | --- | --- |
| Synchronisation message | Gate 1 | Gate 2 |
| 128 | LOW | LOW |
| 129 | LOW | HIGH |
| 130 | HIGH | LOW |
| 131 | HIGH | HIGH |

Table : Synchronisation setting encoding gate status.

The Arduino sketch can be found in TrackSPM\_Alpha. The receiver sketch is CarReceiver.

Two output LED pins are included in the diagram to allow status for the board to be displayed in real time.

* LED 1 on pin 5 Arduino: Voltage from the sound pressure meter causing a reading out of range.
* LED 2 on pin 6 Arduino: Voltage from either of the gates is causing a reading out of range (NOTE NEED TO ADJUST CODE TO ENSURE CORRECT WORKING).

### Setup network for transmitter radio

Open XCTU. Attach one XBee s2c programmable to the USB adapter and plug in.

[You need to help xctu to discover the devices for the first time. Click on add a radio module, manually choose the com port, put in the parameters below and say the radio module is programmable. Then press ok and get lots of reset warnings. You can cancel and be able to find it as below.]

On XCTU discover, choose COM port and default parameters

* Device is discovered at 115200/8/N/1/N API1
* MAC address is 0013A2004147A025
* **This is going to be the on track sound pressure meter:**

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Explanation |
| NI Node identifier | TrackSPM\_Alpha |  |
| BD Baud rate | ~~115200~~ 9600 | Slower speed experiment |
| NB | 0 |  |
| SB | 0 |  |
| ID Pan ID | 8722 | Random but unique |
| CE Coordinator enable | 1 |  |
| SH | 13A200 | Just note these |
| SL | 4147A025 | Just note these |
| DH | 13A200 |  |
| DL | 4147A01F |  |
| AP | 0 | Disables api, puts into transparent mode |

Table : XBee transmitter settings

### Battery booster power pack to on track transmitter cable

The battery booster has a number of cigarette lighter sockets. The cable connecting the booster pack to the on track transmitter is a 12V DC male cigarette socket to LEMO male latch connector. A regulator is located inside the enclosure to bring the voltage down to 9V for the Arduino and shield board. A power switch is located on the enclosure.

Cable length approximately 1.2m.

### Telemetry radios using plan A: In car setup

Roof mounted antenna, with Xbee S2c mounted on an Arduino shield and Arduino. This receives a transmission and obtains a single 15 bit number. This is then converted into a voltage output using a DAC. The Arduino has a Wireless proto shield attached, with XBee S2C connected (to a magnetic antenna). There is a DAC MPC4725 which is linked through the I2C connection straight to the Arduino. The exterior of the box has the following:

|  |  |
| --- | --- |
| Equipment | Specification |
| 12V DC input. | Female Lemo latch connector (2 wire). |
| Power switch. | On / off. |
| LED1 | Simple led |
| LED2 | Simple led |
| LED3 | Simple led |
| Output voltage for gates | Female Lemo (3 wire) (gate 1, gate 2 gnd) |
|  |  |
| Output voltage for sound pressure meter | Female Lemo (2 wire) (output and gnd) |
| Removable antenna connection | Female Lemo or RPSMA |

Table : In car receiver box connections

The schematic of the circuit inside the box is as follows.

The output of the receiver unit is assumed to be 0-5V DC which can be recalibrated to show the same sound pressure level as the input to the transmitter box.

The receiver XBee obtains each byte from the transmitter and stores them in an array. If it gets a byte, it then checks whether a complete message has been obtained (four consecutive expected bytes). If it has, then it converts the two middle bytes (low and high) back into a 16 bit integer (actually 15 bit) and calculates the original voltage seen on the transmitter. The 16 bit (actually 15 bit) can then be mapped to a 12 bit range.

The DAC therefore can have 4096 values with 0 corresponding to 0V output (GND) or 4095 being full scale (VDD). Voltage increments are 1/4096V. As there is nothing on the A0 channel, the address for the device is at 0x62.

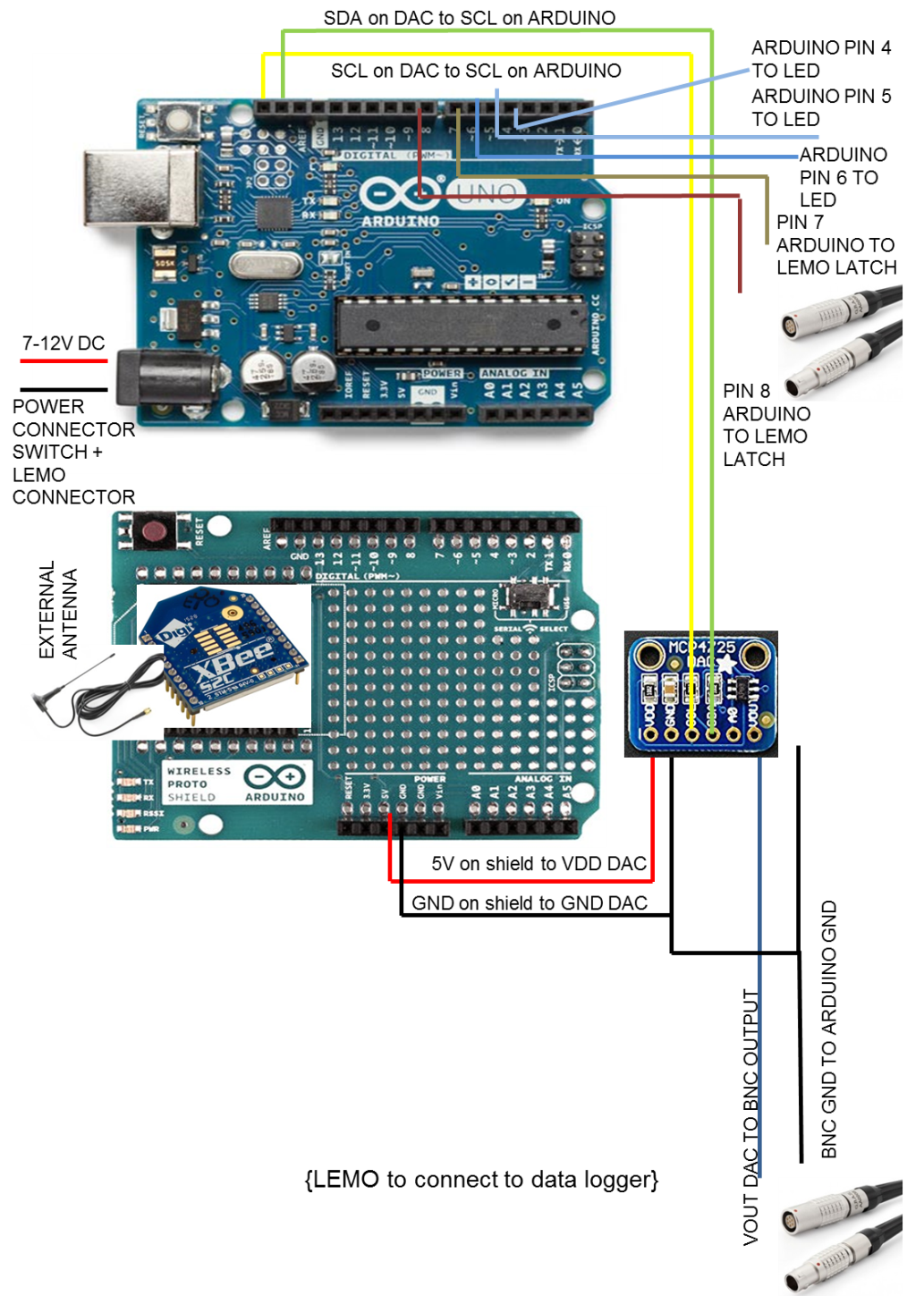


Figure : In car receiver box internals

### Setup network for receiver radio

On XCTU discover, choose COM port and default parameters

* Device is discovered at 115200/8/N/1/N API1
* MAC address is 0013A2004147A01F
* **This is going to be the in car receiver:**

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Explanation |
| NI Node identifier | CarReceiver |  |
| BD Baud rate | ~~115200~~ 9600 | Slower speed debugging |
| NB | 0 |  |
| SB | 0 |  |
| ID Pan ID | 8722 | Random but unique |
| CE Coordinator enable | 0 |  |
| SH | 13A200 | Just note these |
| SL | 4147A01F | Just note these |
| DH | 13A200 |  |
| DL | 4147A025 |  |
| AP | 0 | Disables api, puts into transparent mode |

Table : In car receiver XBee settings

* Open both consoles
* Put in +++ wait for a response ok
* You should see the transmitter sending data to the receiver.

## Equipment connection diagram using plan B

# Range against errors generated

QUESTION WHETHER THERE IS A RANGE VS TRANSMISSION EFFECTIVENESS TO BE ASSERTAINED

# To be completed

Pending (transmitter):

* Sound pressure meter ADC voltage appears correct LED
* Car receiver is transmitting an ok received message sporadically LED
* Gate acquisitions and sending to the receiver
* Speed vs range assessment. Reliability vs speed.
* Antenna choice.
* Checking that the ADC works as anticipated.

Pending (receiver)

* Sound pressure meter DAC voltage appears correct LED
* Track transmitter is ok and we are receiving good data
* Processing of gate status instructions
* Speed vs range assessment. Reliability vs speed.
* Antenna choice and impact on vehicle (aluminium body).
* Calibration method.
* Checking that the DAC works as anticipated.

Look for IP67 or IPX7 (not dust but is water) router and national instruments Ethernet equipment.

Battery for the wireless router – Yuasa

# amp hour batteries + wireless router + antenna