CanSat
Workshop 2016

.uk esero

Tautvydas Mickus, Matthew Rowlings

CanSat Workshop 2016

23-24 September 2016, York, STEM

Approach

Contents

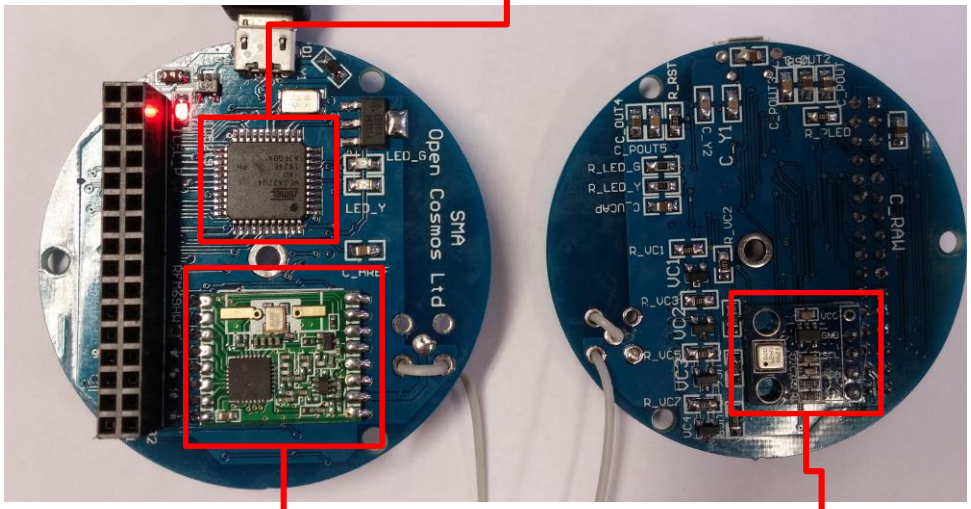
- Introduction to qbcan and whats on-board.
- Setting up Arduino on your PC
- Making LEDs blink and ‘greeting the world’
- qbcan library, variables, reading sensors and displaying data locally
- Introduction to Transceiver (Radio module)
- Sending sensor data wirelessly.

Additionally:

- Minimal soldering and measuring analogue voltage across a solar panel
- Simple saving data to a file and opening in Excel

qbcan compact

Arduino Pro Micro compatible microcontroller



BMP180 Temperature and Pressure sensor

RFM69HW 433MHz Transceiver

qbcan compact

- Microcontroller
  - Fairly powerful, 16MHz ATmega32U4
  - 32 kB of flash memory
  - Unlikely but beware of warning signs in Arduino if memory is running out
- Transceiver
  - RFM69HW powerful 400m+ range transceivers using whip antennas
  - 433MHz with number of different channels
  - Used through qbcan supplied libraries
- BMP180
  - Digital temperature and pressure sensor
  - Used through qbcan supplied libraries

Introduction to Arduino

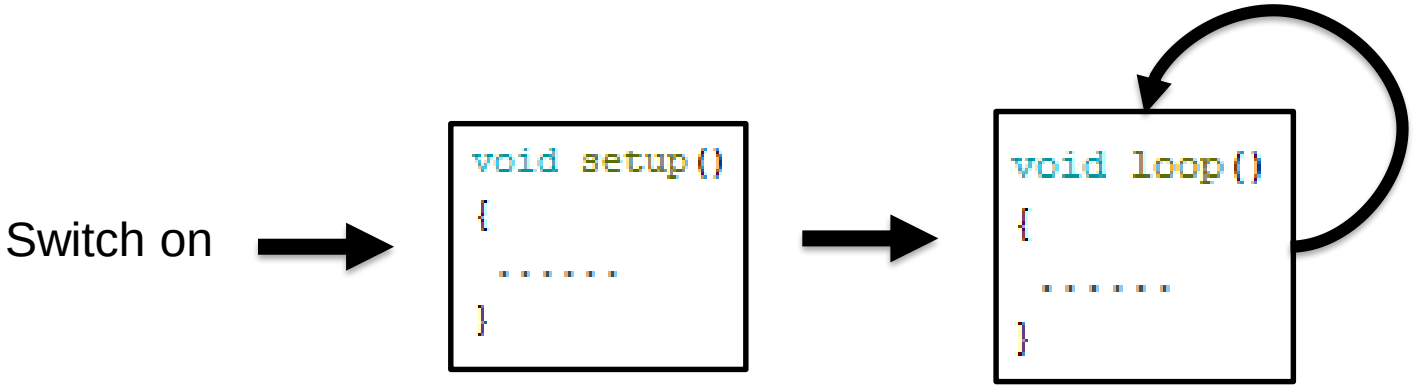
"Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects."

- Setting up Arduino, installing drivers and qbcan library:
  Follow steps for software installation at:
  http://doc.open-cosmos.com/Qbcan_software_installation

  Same website also has quick introduction to programming qbcans in the "Library" section but we will do everything from scratch with more detail

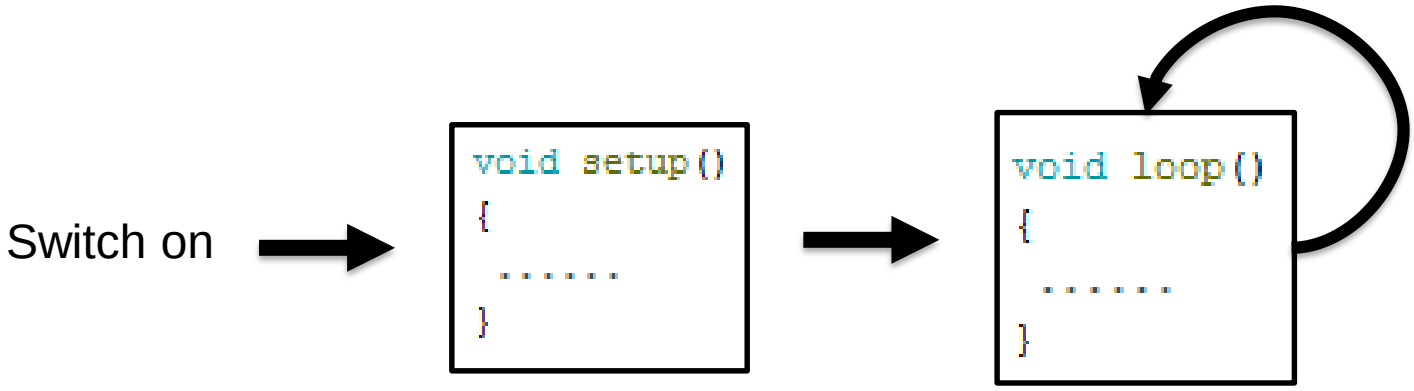
Introduction to Arduino

- C programming language
- Procedural programming
- Simplified via Arduino to only setup() and loop() functions
  - Code inside setup is called only once, when node switches on
  - Code inside loop is repeated



- Huge amount of libraries online for various purposes

LED blinking

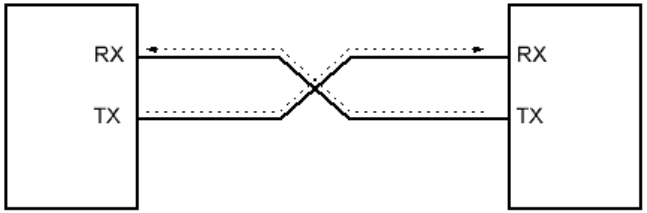


Initialise digital pin as an output

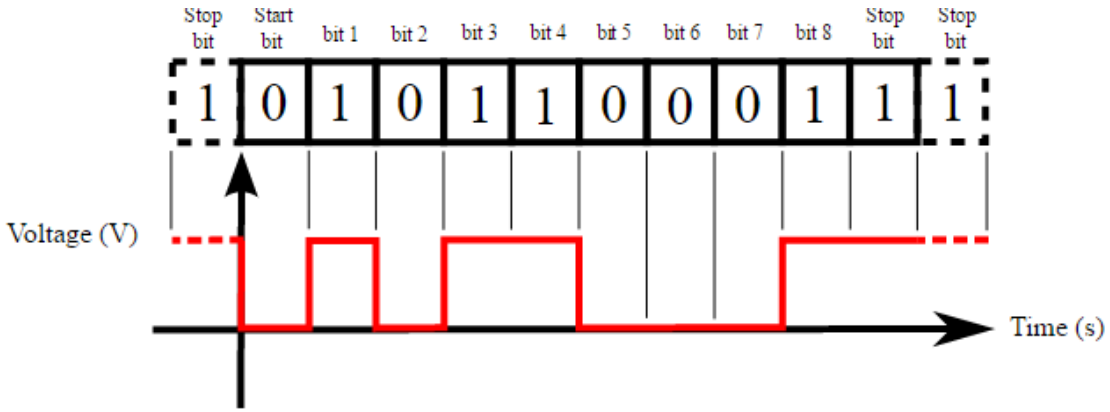
Make the LED flash by setting the pin high/low

Serial Interface (UART)

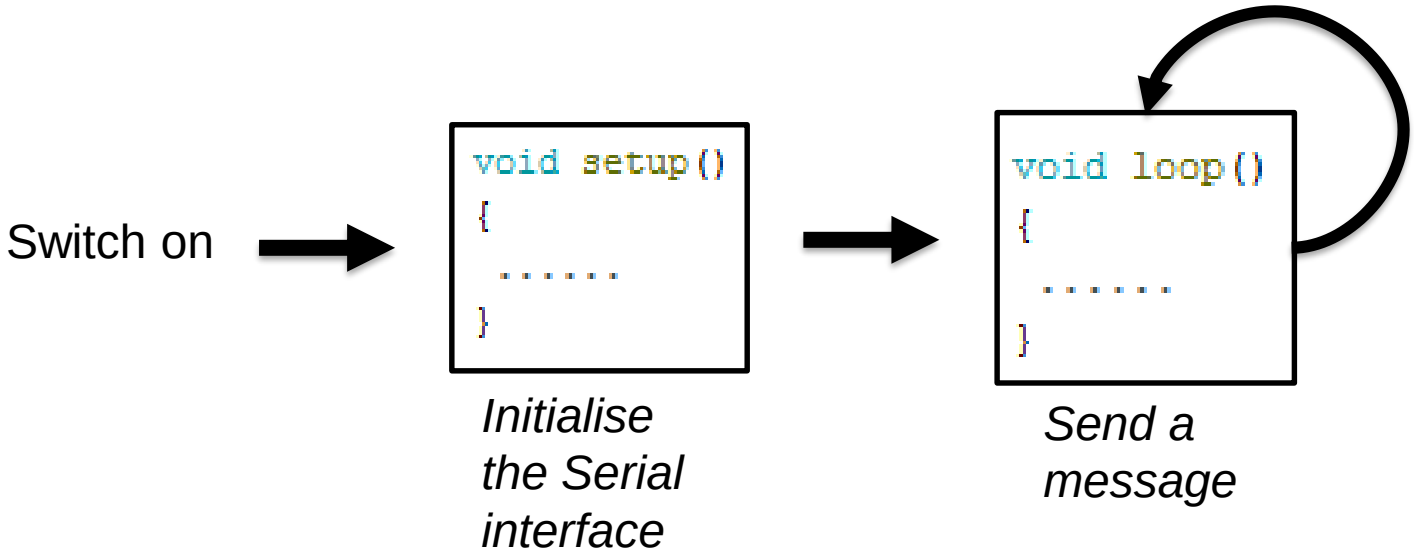
- Often used for debugging
- Probably simplest method of sending data between 2 devices Can be used as one-way link but usually 2 pins on each side (TX, RX)
- Common ground must be available for reference voltage
- Direct digital representation of each character is sent (by default 8bits per character/symbol)



Baud rate – symbol rate
Default usually 9600 baud



Serial communication



Task

- Play around with blinking LED
- Send a message to PC via the Serial interface

- Use “cheat sheet” for summary of the required commands.

Communicating Wirelessly

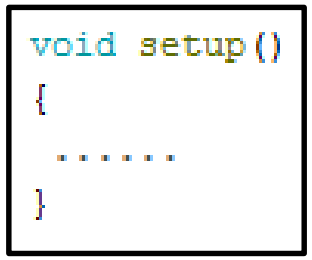
RFM69HW module:

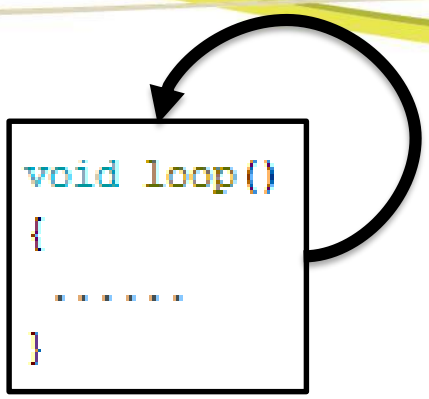
- Controlled via qbcan library
- Just like BMP180 needs initialisation
- Initialisation
  - Need to set channel, node, encrypt key during initialisation
  - Need to specify receiver ID when transmitting data
- Payload/Message is passed as a string (character array)

- Microcontroller on CanSat will just pass the message to the transceiver and tell it to send it to specified node (ground station).
- Microcontroller on Ground will ask if transceiver received anything and once received will obtain the message.
- Transceiver will take care of all the "*magic*" in between.

Communicating Wirelessly

Switch on





Initialise transceiver:

1. *Initialise frequency, channel and ID*
2. *Set to high power (optional)*
3. *Set encryption key (must be same on Ground station)*

Send a message

..and delay a bit.

Sending simple message

```
void loop()
{
  radio.send(1, "Hey", 3); //Send "Hey" to node 1. 3 characters.
  Serial.println("Send complete"); //Display debug message through USB COM port
  delay(500);
}
```

The payload length is important. Longer specified payload length will add confusing characters. Shorter specified length will cut your message.

Sending data: Method 1

```
char payload[50];
void loop()
{
  float pi=3.1415;
  sprintf(payload,"I want to send this number: %d", (int)pi);
  Serial.print(payload);
  radio.send(1, payload, 50);
  Serial.println("Send complete");
  delay(500);
}
```

sprintf is used to construct a string (an array of characters)

- Can only incorporate integers directly.

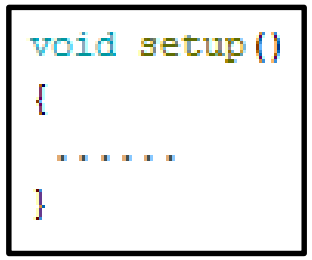
Sending data: Method 2

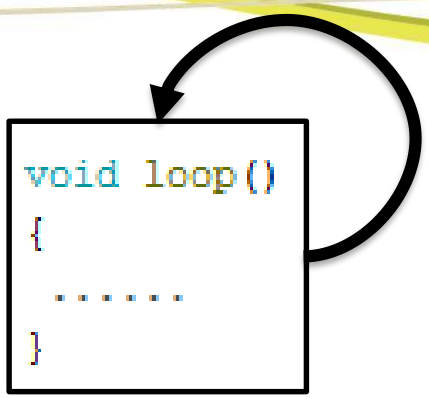
```
char payload[50];
void loop()
{
  float pi=3.1415;
  String payloadstr; //create a payloadstr

  payloadstr+=String("Pi is ");  // build a message "Pi is 3.141"
  payloadstr+=String(pi,3); // 3 specifies the number of decimal places

  payloadstr.toCharArray(payload, 50); //convert this fancy String to a character array that transceiver und

  Serial.println(payload); //debug message
  radio.send(1, payload, 50);
  Serial.println("Send complete");
  delay(500);
}
```

Uses newer type variables and allows incorporating floats easily and might be easier to understand the construction of message

Ground Station

Switch on





Initialise transceiver:

1. *Initialise frequency, channel and ID*
2. *Set to high power (optional)*
3. *Set encryption key (must be same on CanSat)*

If message received – display it on Serial

Receiving data: Method 1

```
void loop()
{
  if (radio.receiveDone())
  {
    Serial.print("Message from Node ");Serial.print(radio.SENDERID, DEC);Serial.print(":");
    for (byte i = 0; i < radio.DATALEN; i++)
    {
      Serial.print((char)radio.DATA[i]);
    }
    Serial.println();
  }
}
```

Loops through received character array and sends it through Serial link character by character

Receiving data: Method 2

```
void loop()
{
  if (radio.receiveDone())
  {

    Serial.print("Message received from Node ");
    Serial.print(radio.SENDERID, DEC);
    Serial.print(":");
    Serial.print((char*)radio.DATA); //points out the whole array of characters at once
    Serial.println();
  }
}
```

Points out the whole array of characters at once rather than looping through.

Task

- Make your CanSat send pressure (and temperature) data to your groundstation
  - Start by sending a simple message to test communications
  - Use cheat sheet ☺

- Use on-screen(projector) “sniffer” if you want to test CanSat before coding Ground station.
  - Switches between channels of each team every 2 seconds

Soldering

Soldering:

1. Clean tip on wet sponge
2. Apply some solder onto the tip
3. Heat up the area/wire you want to solder by touching it with soldering iron
4. Apply solder touching both soldering iron and the area/wire

   Or put some solder on a wire and on area separately and then heat them up while touching together

Tasks:

- Soldering a header onto battery connector wires
- Soldering wires onto a solar panel

Using Analogue to Digital converter

- Common method of sensor measurements
- Converts analogue voltage to digital representation
  - Returns an integer number between 0-1023 (0-5V)
  - Microcontroller is able to sample with 8bit resolution (2^8 different levels)
    - Represents 5V/1023 ~= 4.89mV steps
    - MeasuredValue*0.004889 ~= approximate voltage on pin

- No commands required for setup()

Using Analogue to Digital converter

- No commands required for setup()

```
void setup()
{
  //Initialize serial connection for debugging
  Serial.begin(9600);
  // Initialize pressure sensor.
}

void loop()
{
  int val = analogRead(A1);
  //Display data
  float voltage = val * (5.0 / 1023.0);
  Serial.print("Solar panel voltage: ");
  Serial.println(voltage);
  delay(500);
}
```