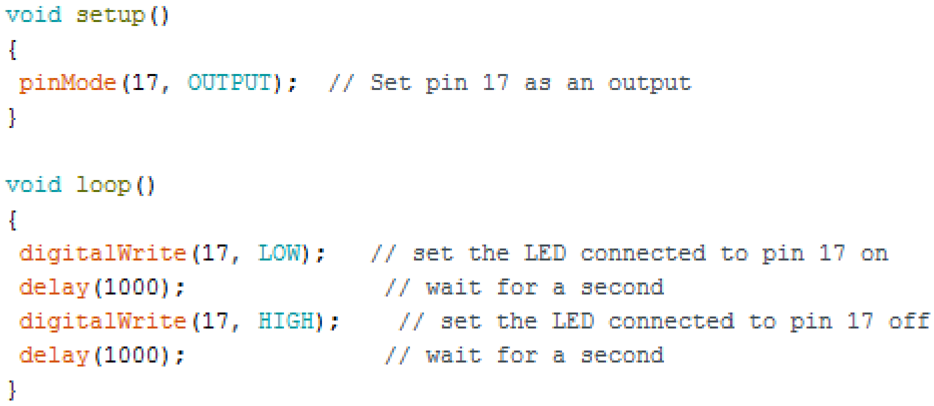
Cheat Sheet – LED blink and Serial

LED blinking Example:



```
void setup()
{
 pinMode(17, OUTPUT);  // Set pin 17 as an output
}

void loop()
{
 digitalWrite(17, LOW);   // set the LED connected to pin 17 on
 delay(1000);              // wait for a second
 digitalWrite(17, HIGH);    // set the LED connected to pin 17 off
 delay(1000);              // wait for a second
}
```

“setup()” part of the code is only run once when module is switched on. “loop()” part of the code is repeated continuously forever. (Unless it breaks ☺)

During setup we need to initialise a pin as an output (LED on the board is connected to pin 17). We do this with “pinMode(***pinNumber,*** OUTPUT)” command.

While the code is running we can switch the pin to High or Low voltage, controlling the LED (on, off). We do this by “digitalWrite(***pinNumber,*** LOW/HIGH)” as shown in example.

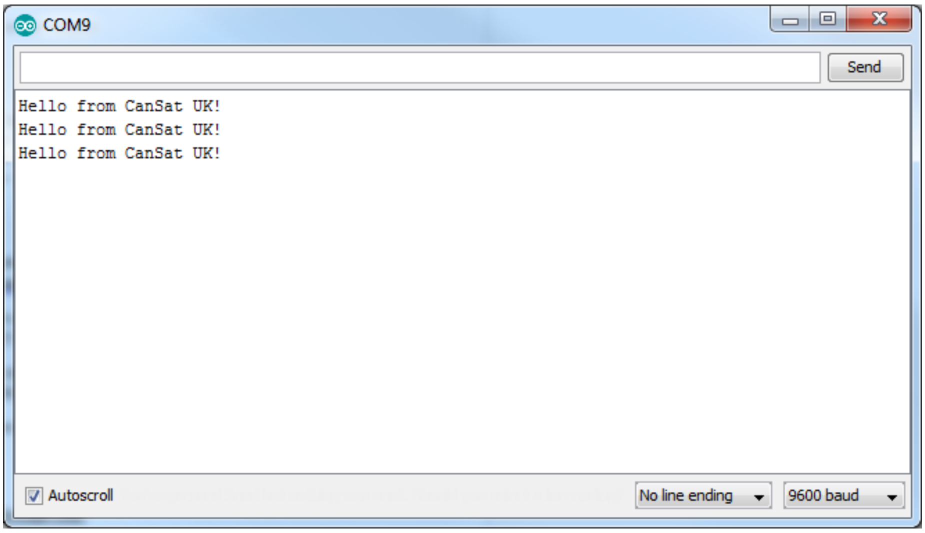
Command “delay(1000)” pauses the program for 1000 milliseconds, therefore making the LED toggle every second.

Example with Serial communications to PC through USB:

```
void setup()
{
 Serial.begin(9600); // Initialise serial interface at specified baud rate.
 pinMode(17, OUTPUT);  // Set pin 17 as an output
}

void loop()
{
 digitalWrite(17, LOW);   // set the LED connected to pin 17 on
 delay(1000);              // wait for a second
 digitalWrite(17, HIGH);    // set the LED connected to pin 17 off
 delay(1000);              // wait for a second
 Serial.println("Hello from CanSat UK!");
}
```

If you upload this code to the module and open Serial Monitor in Arduino you will see message appearing:



```
Hello from CanSat UK!
Hello from CanSat UK!
Hello from CanSat UK!
```

☑ Autoscroll   No line ending   9600 baud

In the setup we use “Serial.begin(***baudRate***)” to initialise the Serial Interface (pretty much like we initialised LED pin as an output). We also specify a baud rate. Make sure that the same baud rate is chosen on Serial Monitor (bottom right corner).

We can then send messages with “Serial.println(***message***)” or “Serial.print(***message***)” commands. Message must be encapsulated in quotation marks.

Serial.println(***message***) – displays message and enters new line so the next message will appear on new line. “Serial.println()” would simply put no text and initiate new line.

Serial.print(***message***) – just displays message.