IoT Coursework Setup Instructions

# Introduction

The coursework involves using an embedded development board to interface with an inertial sensor. The dev board can then communicate wirelessly with an Android app, which you will develop.

The rest of this document describes the setup required to run a reference program, which has been provided for you and has the following functionality:

* Flashes an LED on the dev board
* Sends debugging information to the PC over serial
* Communicates with the IMU breakout board
* Streams sensor data over BLE

It is important to reproduce this behaviour before starting to modify the code running on the dev board, as it will rule out basic problems early on.

# NRF52-DK

You have been provided with a Nordic NRF52-DK board, which we will use with the mbed embedded development platform.

## Flashing the board

First make sure that you can run a basic program on the dev board. You can always return to this example later to verify that your board is still working.

1. Download the **BLE\_LED\_NRF52\_DK.hex** firmware image.
2. Switch on the dev board and connect it to your PC via USB. It should appear as a mass storage device.
3. Copy the firmware to the mass storage device.
4. It should then disconnect from your PC and start running the program.
5. LED1 on the dev board should start to flash once per second.

If this example fails to run you may need to reflash the bootloader on your dev board.

**Note**: We had problems with the bootloader provided on the mbed website, so we don’t recommend that you use this.

### Reflashing the bootloader

1. Download the **J-Link OB-SAM3U128-V2-NordicSemi 160212.bin** bootloader image.
2. Switch off the dev board.
3. Connect to your PC via USB.
4. Whilst holding down the rest button, turn on the dev board.
5. It should appear as a mass storage device.
6. Copy the bootloader image to the MSD.
7. An LED should on the dev board should blink rapidly.
8. Turn the board off and on again.
9. It should now be in a state where you can flash a program file.

# MPU-9250

The InvenSense MPU-9250 is an IMU motion tracking board containing a 3-axis accelerometer, gyroscope and magnetometer. These sensors can be used together to provide full 3D motion tracking.

## Wiring

You have been provided with an MPU-9250, which is mounted on a Sparkfun breakout board. This can be connected to the dev board using the I2C interface. You should make the following connections using jumper cables:

|  |  |  |
| --- | --- | --- |
| **Connection** | **Dev board** | **MPU breakout board** |
| Power | VDD | VDD |
| Ground | GND | GND |
| I2C Clock | P0.27 | SCL |
| I2C Data | P0.26 | SDA |

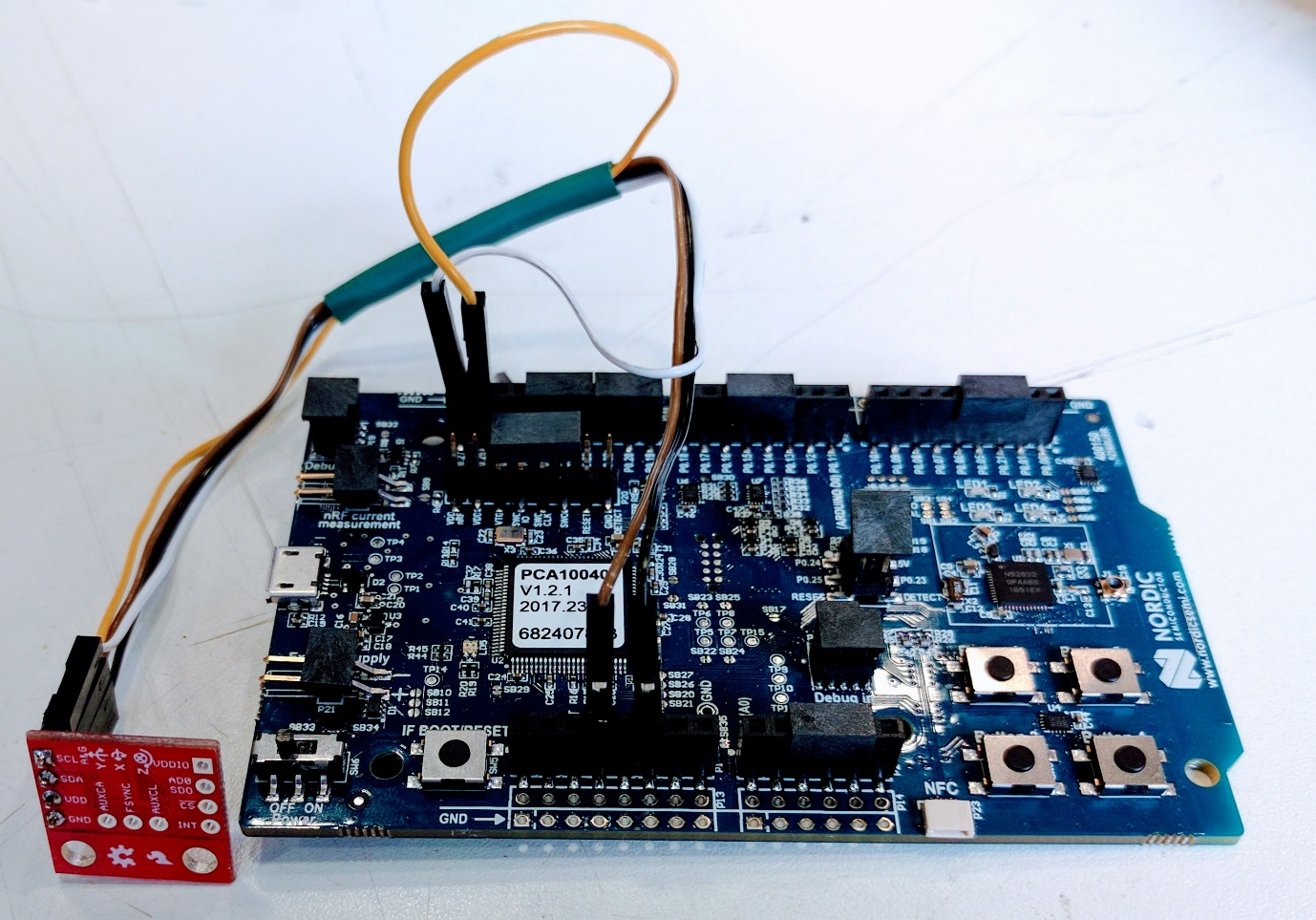


Figure : Connecting the MPU breakout board to the NRF52-DK

## Testing communication

The provided **I2C\_HelloWorld\_Mbed\_NRF52\_DK.hex** program tests communication between the NRF52-DK and the PC (via USB-serial port) and the MPU-9250 (via I2C).

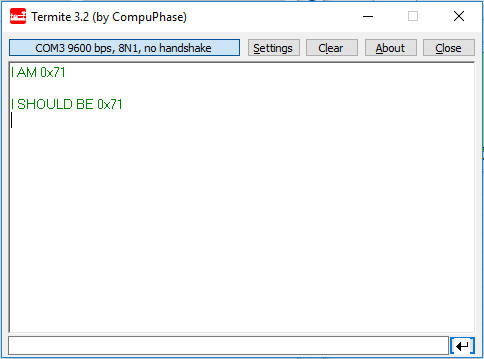
1. Connect the dev board to the PC via USB and switch on.
2. Using a terminal application, open a connection to the J-Link CDC serial port using the following settings:
   1. Baud rate: 9600
   2. 8 bits
   3. Stop bits: 1
3. Copy the above .hex file to the NRF52-DK mass storage device.
4. The program should now run and query the “who am I” address of the MPU.

Figure : Expected serial output when testing MPU comms

# Mbed development platform

We will use the mbed development platform to compile firmware to run on the NEF52-DK board, which is fully supported in mbed OS 5. See the handbook for more information: <https://docs.mbed.com/docs/mbed-os-handbook/en/latest/>

## Online compiler

We recommend that you use the mbed online IDE and compiler for your firmware development. Documentation can be found here: <https://developer.mbed.org/handbook/mbed-Compiler>

Once you’ve registered for an account you’ll have access to your own workspace. This will allow you to compile code and download the resulting firmware image to copy to the dev board.

## Local toolchain

Although it is possible to set up an embedded toolchain on your local machine, this can be tricky. You are free to attempt this at your own risk, but we won’t be providing any support!

## Libraries

There are several mbed libraries for communicating with the MPU-9250, which are available from the online compiler. The reference firmware uses this one: <https://developer.mbed.org/teams/Edutech/code/MPU9250/>

# Android

You will be developing an android application to communicate with the NRF52-DK over BLE. Documentation is here: <https://developer.android.com>

## Testing BLE communication

We recommend using the excellent Nordic nRF Connect app for BLE debugging. This allows you to see the services provided by your BLE device, stream data from notifications and log it to a file. <https://www.nordicsemi.com/eng/Products/Nordic-mobile-Apps/nRF-Connect-for-mobile-previously-called-nRF-Master-Control-Panel>

Try loading the supplied **mbed-os-example-ble-BatteryLevel\_NRF52\_DK.hex** BLE example. This will send fake battery level values.

1. Copy the .hex firmware image to your dev board as described earlier.
2. The dev board should now be discoverable over BLE, with the name “BATTERY”.
3. Open *nRF Connect* on your phone and scan for devices.
4. Press the *connect* button for the “BATTERY” device.
5. Expand the *Battery* service and click the multiple down arrow icon next to *battery* level to stream battery percentage notifications.

