

DUBLIN CITY UNIVERSITY

ELECTRONIC AND COMPUTER ENGINEERING

EE513 Connected Embedded Systems

Assignment 1



Author

Michael Lenehan michael.lenehan4@mail.dcu.ie

Student Number: 15410402

09/03/2020

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the DCU Academic Integrity and Plagiarism at https://www4.dcu.ie/sites/default/files/policy/1%20-%20integrity_and_plagiarism_ovpaa_v3.pdf and IEEE referencing guidelines found at <https://loop.dcu.ie/mod/url/view.php?id=448779>.

Signed: _____

Date: 09/03/2020

Michael Lenehan

Title

Subtitle

Michael Lenehan

Abstract

Contents

1	Introduction	3
2	Assignment Setup	3
2.1	RTC Circuit	3
2.2	I2C Discovery and Testing	4
3	C++ Code	5
3.1	Reading Time and Date	5
3.2	Reading Temperature	5
3.3	Setting Time and Date	5
3.4	Setting Alarms and Interrupts	5
3.5	Novel Functionality	5
4	Linux Kernel Module	5
5	Conclusion	5

1 Introduction

2 Assignment Setup

2.1 RTC Circuit

The following components are used for this assignment:

1. Raspberry Pi 3 Model B+
2. Samsung SD Card
3. Maxim Integrated DS3231 Real Time Clock Breakout Board.

The I2C protocol is used for communications between the RTC and the Raspberry Pi. As such, the RTC must be connected to the Raspberry Pi via the I2C data and clock connections, on pins 3 and 5 (GPIO 2 and 3) respectively. The RTC breakout board operates on a 3.3V supply voltage, and as such the 3.3V (pin 1), and ground (pin 6) pins must also be connected.

Connecting these pins results in the following layout:

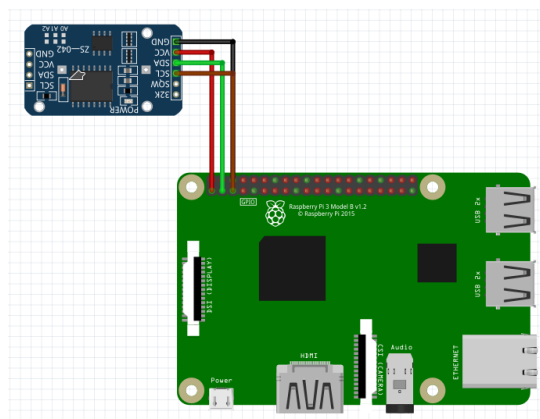


Figure 1: I2C connection from RTC to Raspberry Pi

The physical circuit appears as follows:

2.2 I2C Discovery and Testing

In order to setup I2C on the Raspberry Pi 3 Model B+, the I2C bus must first be enabled. This is achieved by executing the command “raspi-config”. Navigating to the BLANK tab, and selecting BLANK to enable the bus.

Once the circuit is connected, the i2ctools package must be installed in order to verify the connection. The package can be installed by executing the following command:

```
sudo apt install i2c-tools
```

Once installed the I2C bus can be scanned using the “i2cdetect command”. The “-l” option lists the installed busses.

```
i2cdetect -l
```

The output of this command corresponds to the I2C bus that the device is connected to. Using the “-r” option, bus 1 can be scanned using the SMBus “receive byte” method. The “-y” option in this case disables user input for confirmation.

```
i2cdetect -y -r 1
```

Finally the “i2cdump” command can be used to dump the contents of the specified address, in this case “0x68” on bus 1:

```
i2cdump -y 1 0x68
```

3 C++ Code

Using the provided C code, a DS3231 class is created, with methods each of the required functions.

3.1 Reading Time and Date

The provided code contains all of the functionality required to connect to the RTC, read and display the current time. These features were initially extracted into methods in order to make the code more modular. To read the time and date, the I2C bus connection must be opened, the device address must be connected to, the read address must be specified, and the values must be read into a buffer. The desired values can then be displayed from the buffer.

In the case of the Raspberry Pi, the RTC is on I2C bus 1. As such, the code for opening this connection is as follows:

3.2 Reading Temperature

3.3 Setting Time and Date

3.4 Setting Alarms and Interrupts

3.5 Novel Functionality

4 Linux Kernel Module

5 Conclusion