NAZARBAYEV UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

Project Assignment #2

ROBT305 – EMBEDDED SYSTEMS

Fall Semester 2018

IMU SENSOR GUI DEVELOPMENT USING QT AND BBB GPIO PROGRAMMING

DUE TIME AND DATE

- **Class presentation and report uploaded to Moodle on Thursday 27 September.**

LEVEL OF COLLABORATION ALLOWED

You will be working in groups of two students

DELIVERABLES REQUIRED

You group is required to prepare and submit to Moodle system an MS Word report with Linux terminal screenshots for all major tasks and steps from this assignment. In addition you will present your assignment in class.

REFERENCES

- **Derek Molloy,"Exploring BeagleBone. Tools and Techniques for Building with Embedded Linux", Wiley, 2015 (available in Moodle and in library)**
- **G. Lazar, R. Penea, Mastering Qt5, Pact, 2016 (available in Moodle)**

INTRODUCTION

In this assignment you will develop an IMU sensor orientation visualization GUI.

ASSIGNMENT DETAILS

TASK 1. INTRODUCTION TO QT GUI DEVELOPMENT (20% OF THE TOTAL GRADE)

In this task you will work on your laptop/PCs.

1. Download and install the community (free) version of the Qt C++ application development environment from http://www.qt.io/

2. Follow the online C++ GUI with Qt Tutorial videos #2 up to #14 (if deemed necessary) from https://www.youtube.com/watch?v=aMUh9DmFLto and learn how to develop C++ GUIs in the Qt Creator program

3. Think of some interesting (not very complicated) GUI project and implement it for the class presentation in your laptop/PC. Examples: calculator, mouse pointer coordinate tracking, etc. The main point of this task is to show you can understand and develop a relatively simple Qt GUI application.

TASK 2. IMU GUI DEVELOPMENT (45% OF THE TOTAL GRADE)

In this task you will develop a client/server application to visualize an IMU sensor orientation on your laptop/PC. The BBB with connected IMU sensor acts as a server and sends processed orientation data to your laptop/PC (client). The client runs a Qt GUI for visualizing the IMU sensor 3D orientation. Please implement the following:

1. Study the Magwick IMU signal fusion algorithm carefully and define in what physical values the algorithm accepts the input sensor data. In your IMU sensor read programs convert the sensor raw measurements to physical values using scaling factors defined based on the maximum measurements values and the above defined output ranges. Write your calculations in the report.
2. Study the principles of client/server application development with BBB in **Chapter 10 of the BBB textbook.** In particular, focus on the **C++ Client/Server section on pp. 412-415.**
3. Study the Qt GUI application development in **Chapter 11 of the textbook.** In particular, focus on the **Fat Client Qt GUI Application on pp. 455-462.**
   Examples program codes from Chapters 10 and 11 are available in Moodle.
4. Based on your Project 1 developments implement on the **server side (BBB)** the code with the following functions:
   1. Reading the IMU raw sensor measurement data and converting it to physical values as explained in Steps 1 and 2.
   2. The open-source Magwick IMU sensor measurement fusion algorithm accepts the IMU data and sends the quaternion estimates to the server socket communication for reading by the client side. **Please make the quaternion component normalization before sending the data.**

   The textbook describes the use of the Apache web server embedded into the BBB Debian image. You may use alternative socket communication servers such as NodeJS, etc. if you prefer.
5. On the **client side (your laptop/PC)** you will implement GUI application in the Qt Creator which handles:
   - the client socket communication. It reads the data.
   - runs the data processing and visualizes IMU orientation.

   Please develop the IMU sensor orientation visualization in the Qt GUI using the provided in Moodle **test-opengl** framework based on OpenGL and QTthread classes. You may use alternative visualization tools, e.g. QtCanvas3D or others, if preferable.

TASK 3. BBB GPIO PROGRAMMING (35% OF THE TOTAL GRADE)

Your group will implement an embedded system consisting of an electronic circuit controlled by a BBB board. In a simple scenario the circuit will include a button and two LEDs that are connected to the BBB board GPIO. The program will monitor the button status and switch between the two LEDs once the button is pressed.

1. Study principles of BBB GPIO interfacing in Chapters 4 and 6 of the textbook and video on http://derekmolloy.ie/beaglebone/beaglebone-gpio-programming-on-arm-embedded-linux/
2. As a basis for the project development you can implement a circuit shown in Figure 1 below. The circuit includes: 1 LED, 1 push button, 1 2N2222A NPN transistor, 2 10kOhm and 2 100Ohm resistors. These components are available in lab 7.327 – check electronic component kits that you may have used in the Electric Circuita course labs in Year 2.
3. Study the example C++ code for the circuit (available in Moodle): **SimpleGPIO.cpp SimpleGPIO.h, TestApplication.cpp** You can transfer the code to the BBB through SSH and compile it in the Ubuntu terminal using

```
g++ -o2 -Wall TestApplication.cpp SimpleGPIO.cpp -o TestApplication
```

4. Add a second LED and implement a circuit with 1 push button and 2 LEDs connected to the BBB board GPIO. Write the program that monitors the button status and switches between the two LEDs once the button is pressed.

GRADING CRITERIA

Demonstration and grading of your working Qt appilication (Task 1), BBB&IMU setups (Task 2) and BBB GPIO (Task 3) setups and reports (with questions/answers) will be done during the class lab session on Thursday 27 September.

Please prepare a detailed report with program code and submit it to the project folder in Moodle by the end of Thursday 27 September.

This project evaluation will be done using individual grading depending on the level of participation and understanding of the project assignments.

Late submission penalty – 10% per day

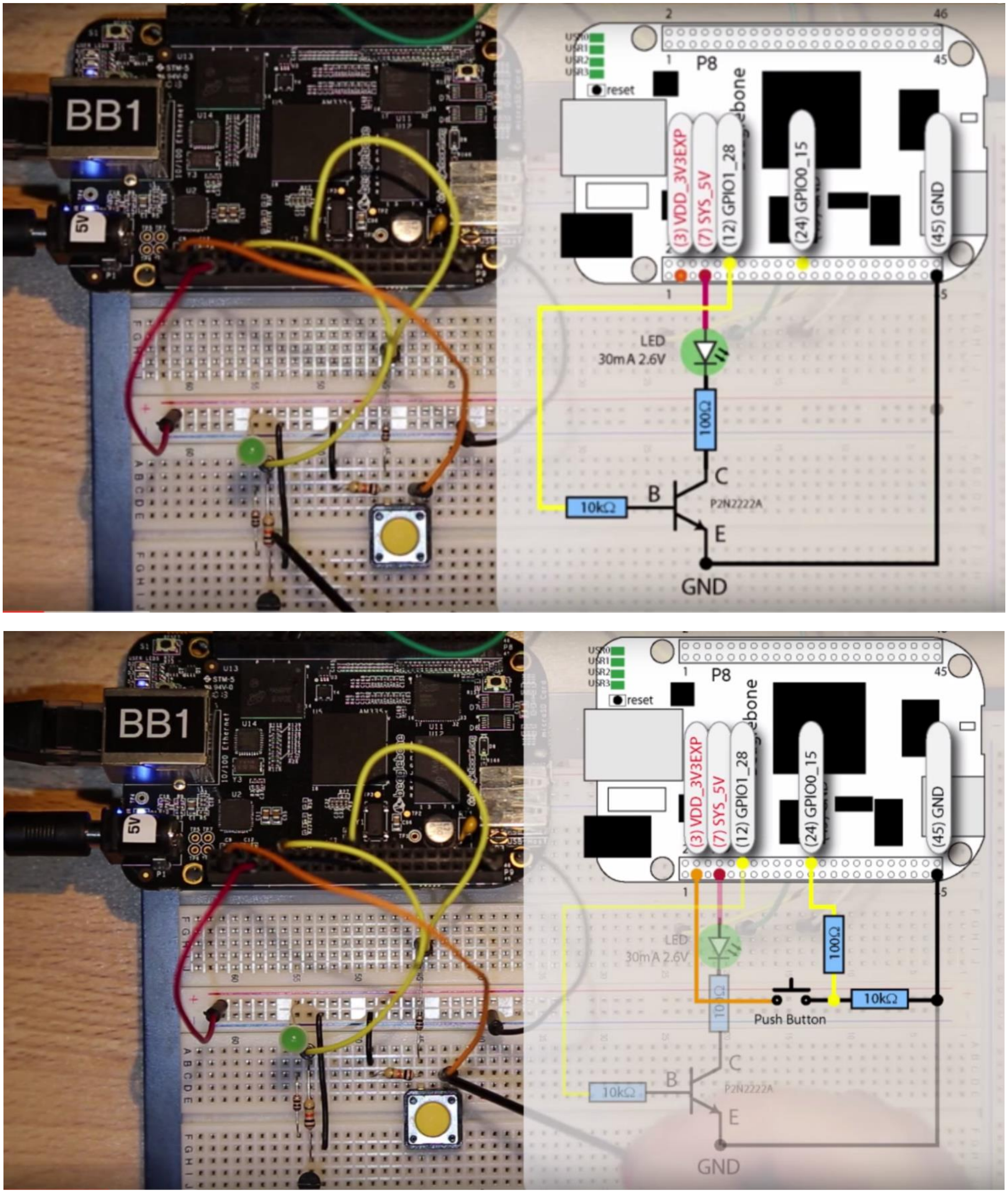

Figure 1 – LED and button connection to BBB using GPIO.