PROJECT SETUP

Introduction

To ease the use of this application during the practical classes and the development of the new or modified application by the University Student or young developers, the template project is provided. The students just need to follow few steps in order to successfully use this application. Therefore, a description of how to download and set up the template project is given first. This step by step walk-through is based on using Qt Application version 4.8.6 on a Raspberry Pi Unix based system.

Hardware Connection

Before doing anything it's important to know the hardware connection between the Raspberry Pi and BITalino board, because without a correctly connection none of the coming step will mean anything. Therefore Figure A.1 shows the connection and the summary description of it is shown in the Table A.1 below.

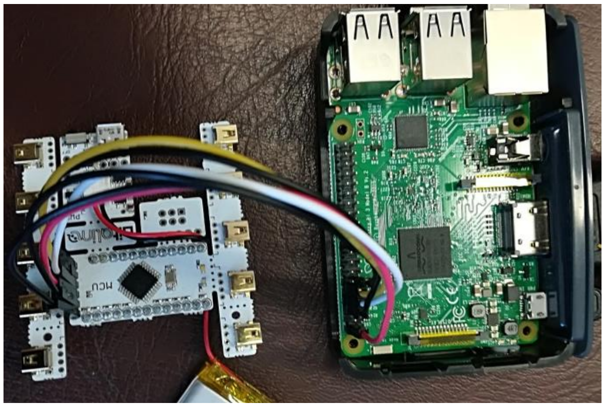

Figure 1: BITalino Board and Raspberry Pi connection.

Raspberry Pi			BITalino Board
Pin Number	**Description**	**Wire**	**Description**
Pin 2	VCC	Red	VCC
Pin 6	GND	Black	GND
Pin 8	Tx	White	Rx
Pin 10	Rx	Yellow	Tx
Pin 11	GND	Black	GND
Pin 20	GND	Black	CTS

Table 1: Raspberry Pi and BITalino Board Pins Connection.

Download the Project

The first thing needed is the template project itself, which is found on GitHub. Download the template project, rename it with the desired name and move it to the Home Pi-folder, as shown in Listing A.1.

```
git clone https://github.com/twahakabika/Data-Acquisition-Tool
mv template <path to Home Pi-folder>/<new_directory_name>
```

Listing 1: Download the Template Project

Project Environment Setup

In order for Raspberry Pi to be ready for running the application, the gcc, g++ software and Qt cross-platform application must be properly installed. The listings below show very specific how to download gcc 4.4 and g++ 4.4 and Qt 4.8.6 because are the versions of the software which were used during the implementation and testing of the developed application.

```
sudo apt-get install gcc4.4
sudo apt-get install g++4.4

sudo apt-get install qt4-default
sudo apt-get install qt4-dev-tools
sudo apt-get install qtcreator
sudo apt-get install xterm
sudo apt-get install git-core
sudo apt-get install subversion
```

Listing 2: Install gcc, g++, Qt and all Dependencies

Also the Raspberry Pi console must be disabled in order to allow the efficient use of the serial communication. This can be done by adding the line *enable_uart = 1* at the bottom of the *config.txt* file, to reach this file, from the terminal type, *sudo nano /boot/config.txt*. This will basically set serial 1 or ttyS0 as Mini UART. But the baud rate of the mini UART is linked to the core frequency of the VPU on the VC4 GPU. This means that as the VPU frequency governor varies the core frequency, the baud rate of the UART also changes. This makes the UART of limited use in the default state. To overcome this challenge *dtoverlay=pi3-miniuart-bt* is added below the *enable_uart = 1* on the *config.txt*. This line switches the Raspberry Pi 3 Bluetooth function to use the mini UART (ttyS0), and restores UART (ttyAMA0) to GPIOs 14 and 15 which increase the maximum usability of baud rate to the GPIOs.

Import Project to Qt Application

When the project is downloaded and move, the next step is to import it to Qt Application. This is done by launching Qt Application and opening the project, as shown in Listing A.4.

```
File->Open->(Select the downloaded folder)
```

Listing 3: Importing template project to Qt Application