Co-funded by the
Tempus Programme
of the European Union

Manipulating with Raspberry Pi

Galyna TABUNSHCHYK

Prof. Software Tools Department

Zaporizhzhia National Technical University

Co-funded by the
Tempus Programme
of the European Union

Plan

1. Zaporizhzhia National Technical University
2. About Myself
3. What is possible to do with Raspberry Pi
4. How to manipulate Raspberry Pi
5. Other projects

Co-funded by the
Tempus Programme
of the European Union

Ukraine

603,000 km2
Over 45 million people

https://www.youtube.com/watch?v=qZMMJo7jOTQ&feature=youtu.be

Co-funded by the
Tempus Programme
of the European Union

Zaporizhzhia National Technical University

- 117 years since the establishment
- 18,000 students of all forms of learning
- 12,000 full-time students
- More than 1,500 faculty and staff
- Bachelor, Master, PhD

Co-funded by the
Tempus Programme
of the European Union

Software Tools Department

Specialties

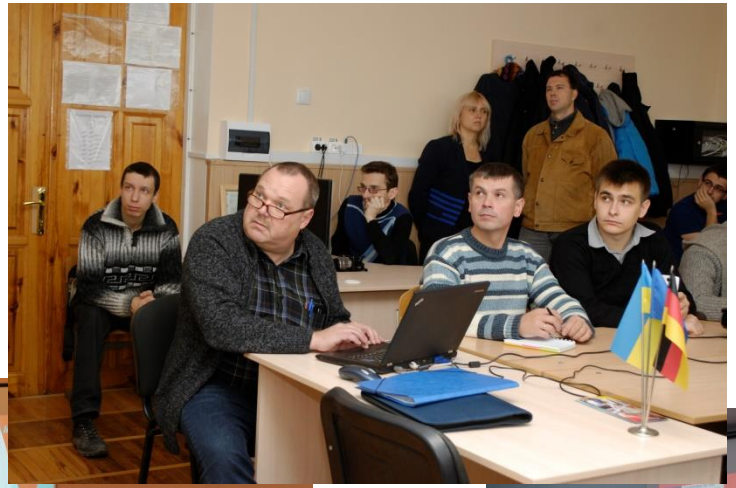
- Engineering of Software;
- Computer science and Information technologies.

Education levels

- Bachelor;
- Master;
- PhD.

Co-funded by the Tempus Programme of the European Union

Guest Lectures



Co-funded by the Tempus Programme of the European Union

Students internship 2016-2020

DAAD
Deutscher Akademischer Austausch Dienst
German Academic Exchange Service

Fachhochschule Dortmund
University of Applied Sciences and Arts

berlin

th

KU LEUVEN

CARINTHIA UNIVERSITY OF APPLIED SCIENCES
FACHHOCHSCHULE KÄRNTEN

Funded by the Erasmus+ Programme of the European Union

Co-funded by the Tempus Programme of the European Un

15-18 October, 2015, VNTU, Vinnyts

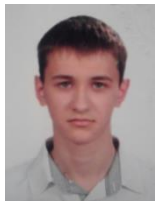
About Myself

- Professor of Software Tools Department of Zaporizhzhya National Technical University, Institute of RadioElectronics and Informatics, Faculty Computer Sciences and Technologies
- supervising work of PhD students;
- **Courses**: Object Oriented Programming, Designing and Modelling of Software in Embedded Systems, Requirements Analysis, Quality of Informational Systems, Software Project Management, Software Quality and Testing;
- Local Project Manager in Tempus Project 544091-TEMPUS-1-2013-1-BE-TEMPUS-JPCR - Desire
- head of scientific research group of Reliability of Informational Systems at Software Tools Department

Co-funded by the
Tempus Programme
of the European Union

ISR Team

- Appear in 2011
- Work:
  - System Verification
  - Planning and Monitoring of Software Development Process
  - Risk Analysis for Industrial Application
  - Reliability of Embedded Systems
  - Video Processing



Co-funded by the
Tempus Programme
of the European Union

Embedded Software Development

Total hours 108h

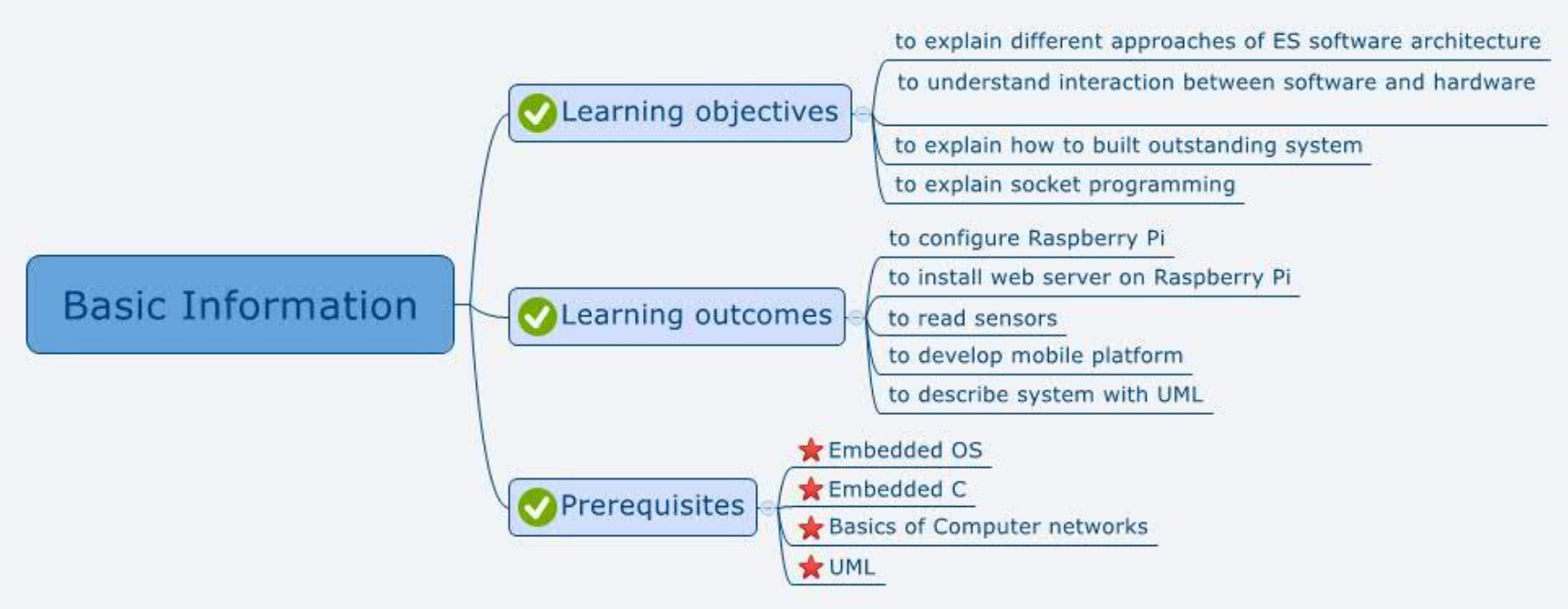
- Lectures: 12 h
- Lab works: 24 h
- Self work 72 h

Lecturer
Galyna TABUNSHCHYK ,
PhD, Prof.
galina.tabunshchik@gmail.com

Teaching Assistant

Natali Myronova
natali.myronova@gmail.com

Eygeniy Tverdokhleb
junta.kristobal@gmail.com

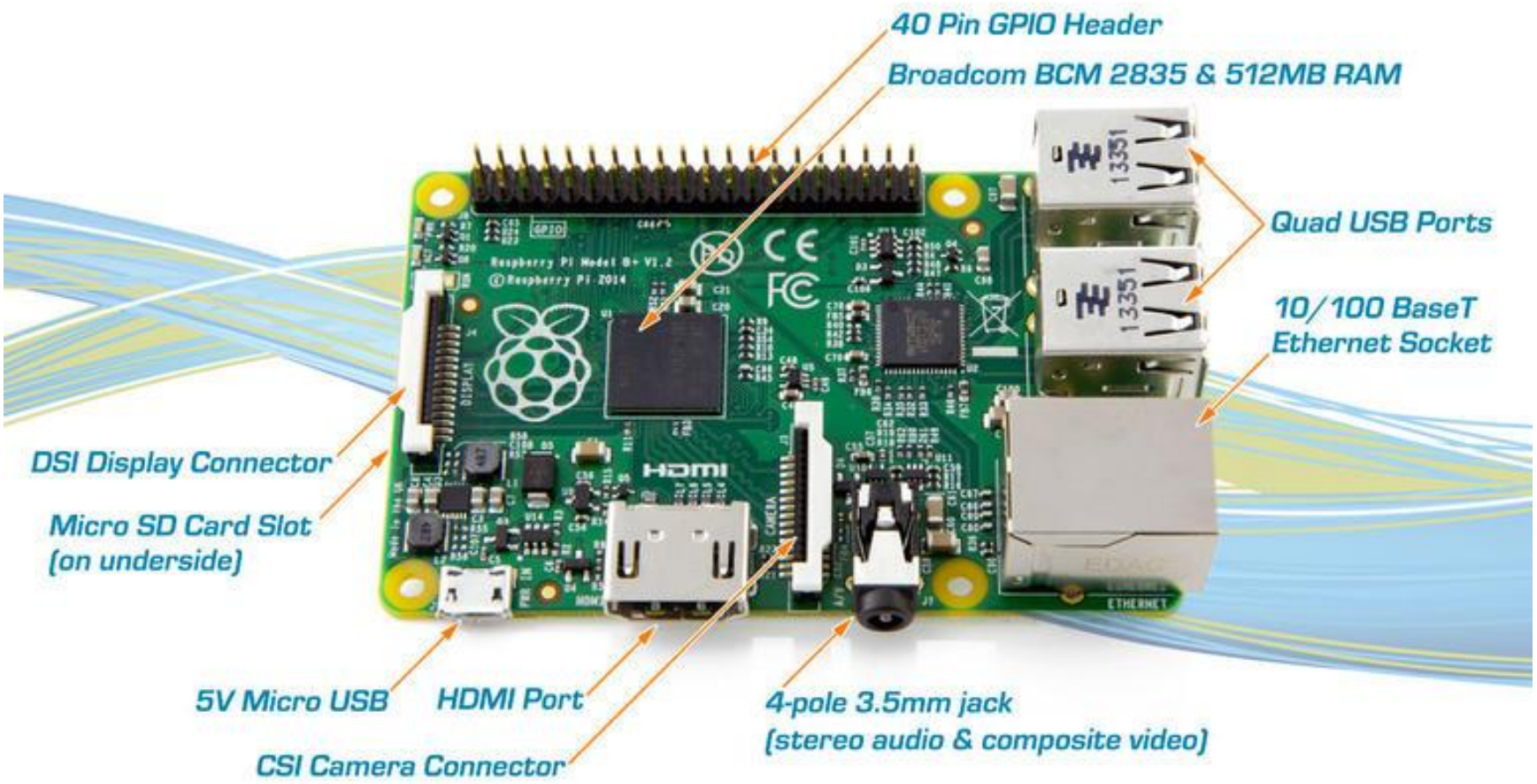

Basic Information
Learning objectives
to explain different approaches of ES software architecture
to understand interaction between software and hardware
to explain how to built outstanding system
to explain socket programming
Learning outcomes
to configure Raspberry Pi
to install web server on Raspberry Pi
to read sensors
to develop mobile platform
to describe system with UML
Prerequisites
Embedded OS
Embedded C
Basics of Computer networks
UML

Week	Subject
1	Introduction
2-3	Modelling of software for Embedded Systems
3-4	Standard component models
5-6	Architecture of the software for Embedded Systems
6-8	Templates for Software Architecture for Embedded Systems
9-10	Socket programming
11-12	Programming Linux Socket

Experiments, Projects, Lab Works	Subject
Lab work 1	Configuring Raspberry Pi
Lab work 2	Installing Web-server at Raspberry Pi
Lab work 3	Developing QT application at Raspberry Pi
Lab work 4	Reading sensors from extension board
Lab work 5	Developing Project on Raspberry Pi

What is Raspberry Pi???

Co-funded by the
Tempus Programme
of the European Union



Co-funded by the Tempus Programme of the European Union

Raspberry Pi

Raspberry Pi:	Model A+	Model B	Model B+	2, Model B
Quick summary:	Cheapest, smallest single board computer.	The original Raspberry Pi.	More USB and GPIO than the B. Ideal choice for schools	Newest, most advanced Raspberry Pi.
Chip:	Broadcom BCM2835			Broadcom BCM2836
Processor:	ARMv6 single core			ARMv7 quad core
Processor Speed:	700 MHz			900 MHz
Voltage and Power Draw:	600mA @ 5V			
GPU:	Dual Core VideoCore IV Multimedia Co-Processor			
Size:	65x56mm	85x56mm		
Memory:	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	1 GB SDRAM @ 400 MHz	
Storage:	Micro SD Card	SD Card	Micro SD Card	
GPIO:	40	26	40	
USB 2.0:	1	2		
Ethernet:	None			
Audio:	Multi-Channel HD Audio over HDMI, Analog Stereo from 3.5mm Headphone Jack			

Co-funded by the
Tempus Programme
of the European Union

Raspberry Pi 3

- **SoC:** Broadcom BCM2837
  CPU: 4× ARM Cortex-A53, 1.2GHz
  GPU: Broadcom VideoCore IV
  RAM: 1GB LPDDR2 (900 MHz)
  Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
  Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
  Storage: microSD
  GPIO: 40-pin header, populated
  Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

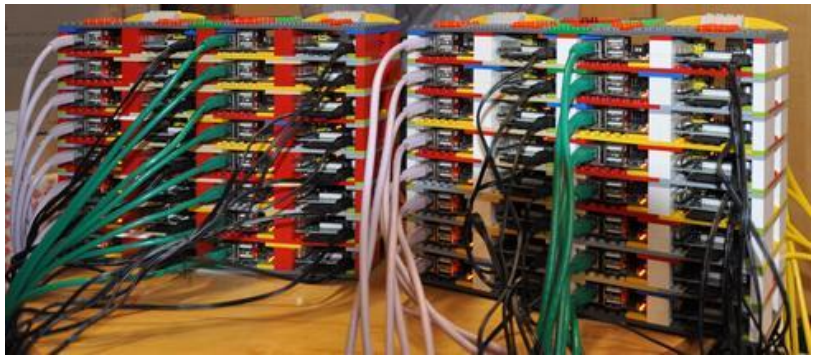
What you can do with Raspberry Pi??????????

- Robotics

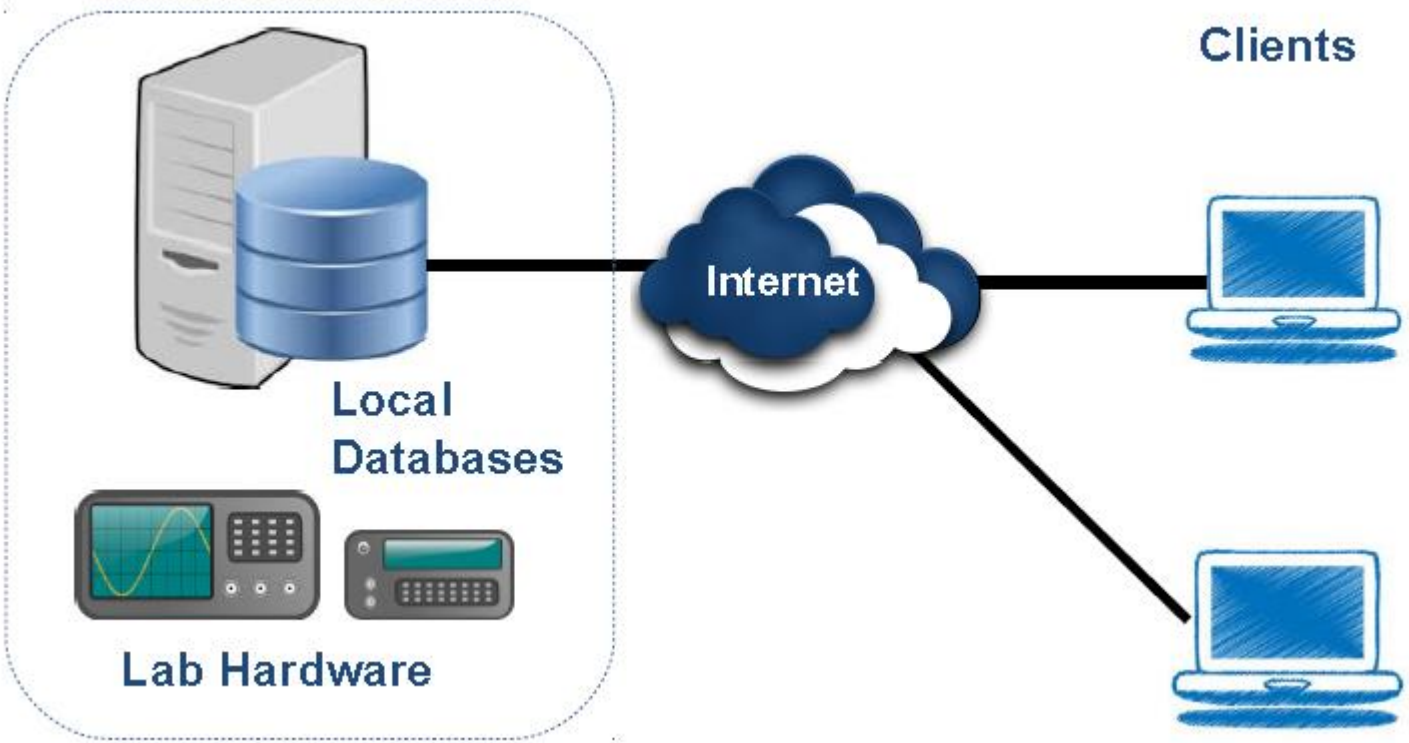
https://www.youtube.com/watch?v=j_1JFnwOFwI

- Learn Programming
  - Scratch
  - C++
  - Python

- Web Server
- Media Server
- Cluster



REMOTE LABORATORIES



Co-funded by the
Tempus Programme
of the European Union

TYPES OF RLABs

Interactive

- Interactive experiments are those in which the user monitors and can control one or more aspects of the experiment during its execution.
- • …require real-time control.
- …are performed in human-time. Longer periods of single user control

Batched

- Batched experiments are those in which the entire course of the experiment can be specified before the experiment begins. Batched experiments should be queued for execution in order to maximize the efficiency of the lab server.

Co-funded by the
Tempus Programme
of the European Union

ACCESS TO RLABs

Calendar

Lab session typically takes longer ● User can reserve an specific time-slot ● Reliable: lab server will be available at reserved time

Queue

Usually implemented as first in first out (FIFO) ● Experiments usually run fast ● Requests can be prioritized

DEVELOPMENT OF THE RLABS

- Design Lab Clients
- Bound by Lab-specific UI requirements
- Design Lab Server
- Bound by lab instrumentation, desired functionality
- Design Client-Server communication framework
- Implement Web Services
- Create/parse experiment specification
- Ensure proper ICT infrastructure
- Ensure proper system security
- Collaboration with IT department
- Lab must be reachable from external network
- Setup of the server environment respecting institution's network policies

Requirements for remote experiments

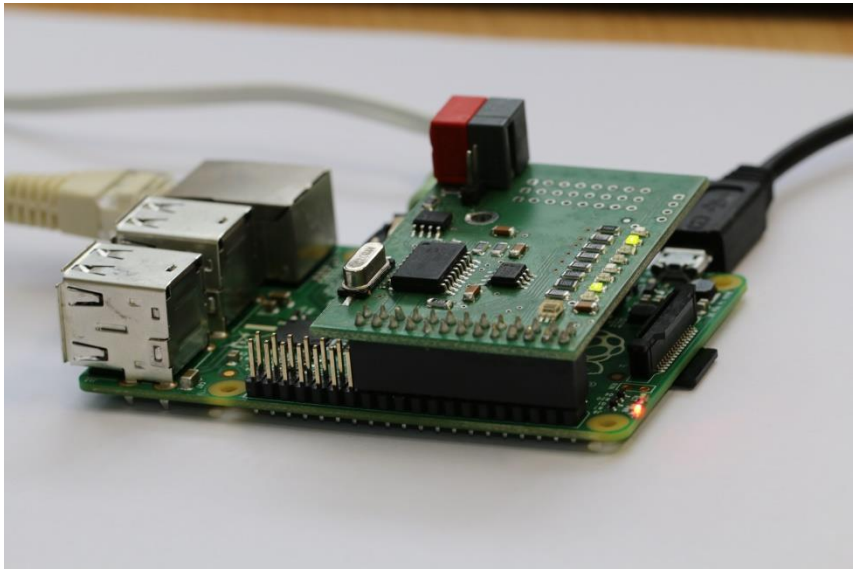
- availability 24/7
- should provide possibility for hardware and software testing
- no requirement for students HW
- should improve students skills in software development

Prerequisites for students

- Basic knowledge in Linux
- C++ skills
- Basic knowledge in Electronic Devices
- Software quality metrics
- Basics in computer systems and network

Co-funded by the
Tempus Programme
of the European Union

New remote experiments



Hardware:

Raspberry Pi Model B

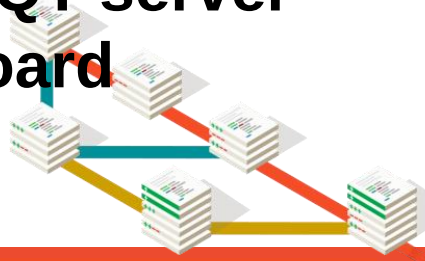
Expansion board

Wifi, BLE4 adapters, webcam

Software:

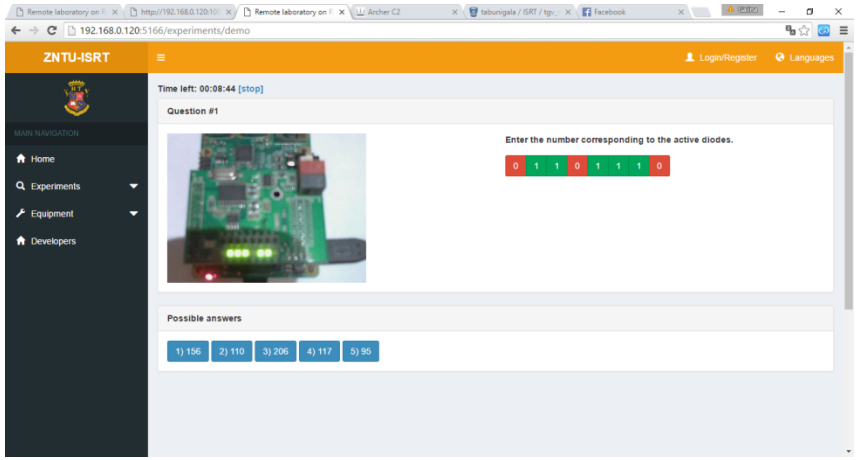
Raspbian Linux, Apache, MySql, C++, git, QT server for expansion board

Pilot usage:
Master course:
Embedded Software Development
Bachelors course:
Design of Informational System



Co-funded by the
Tempus Programme
of the European Union

Two demo Modes

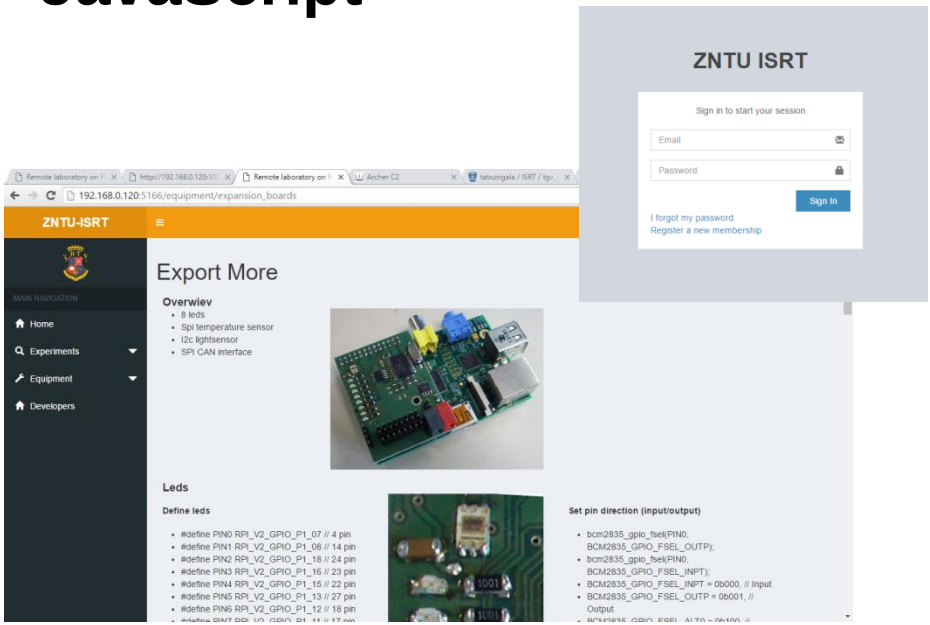


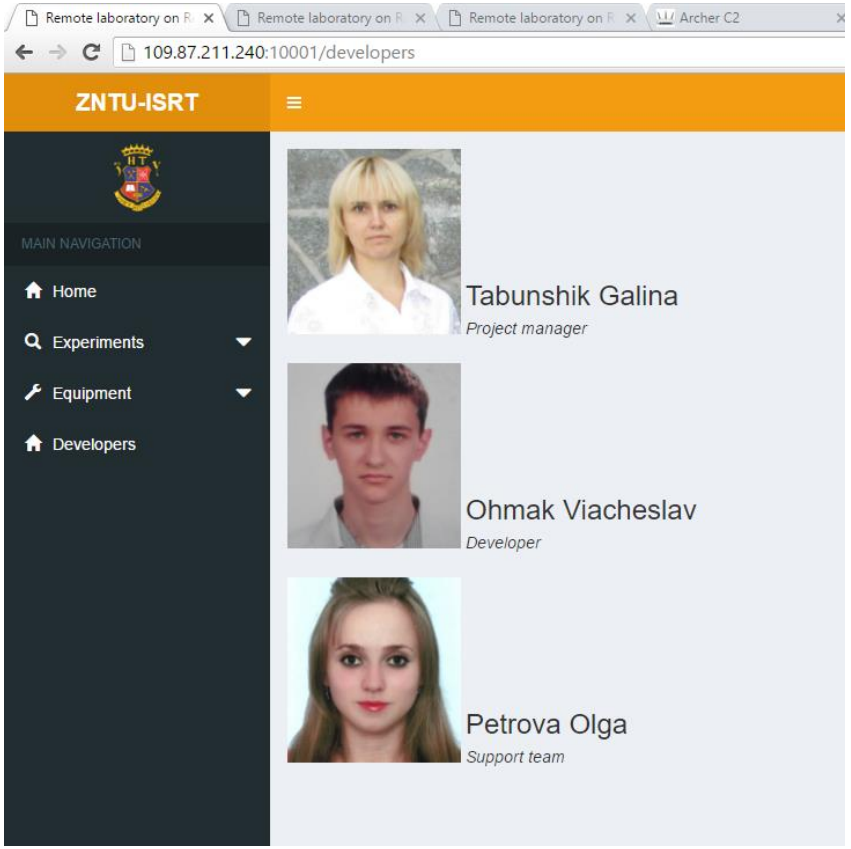
- Manipulating with leds on Thomas More expansion board with C++
- Manipulation with step engine and light sensors by Python and C++

Co-funded by the
Tempus Programme
of the European Union

Web-server

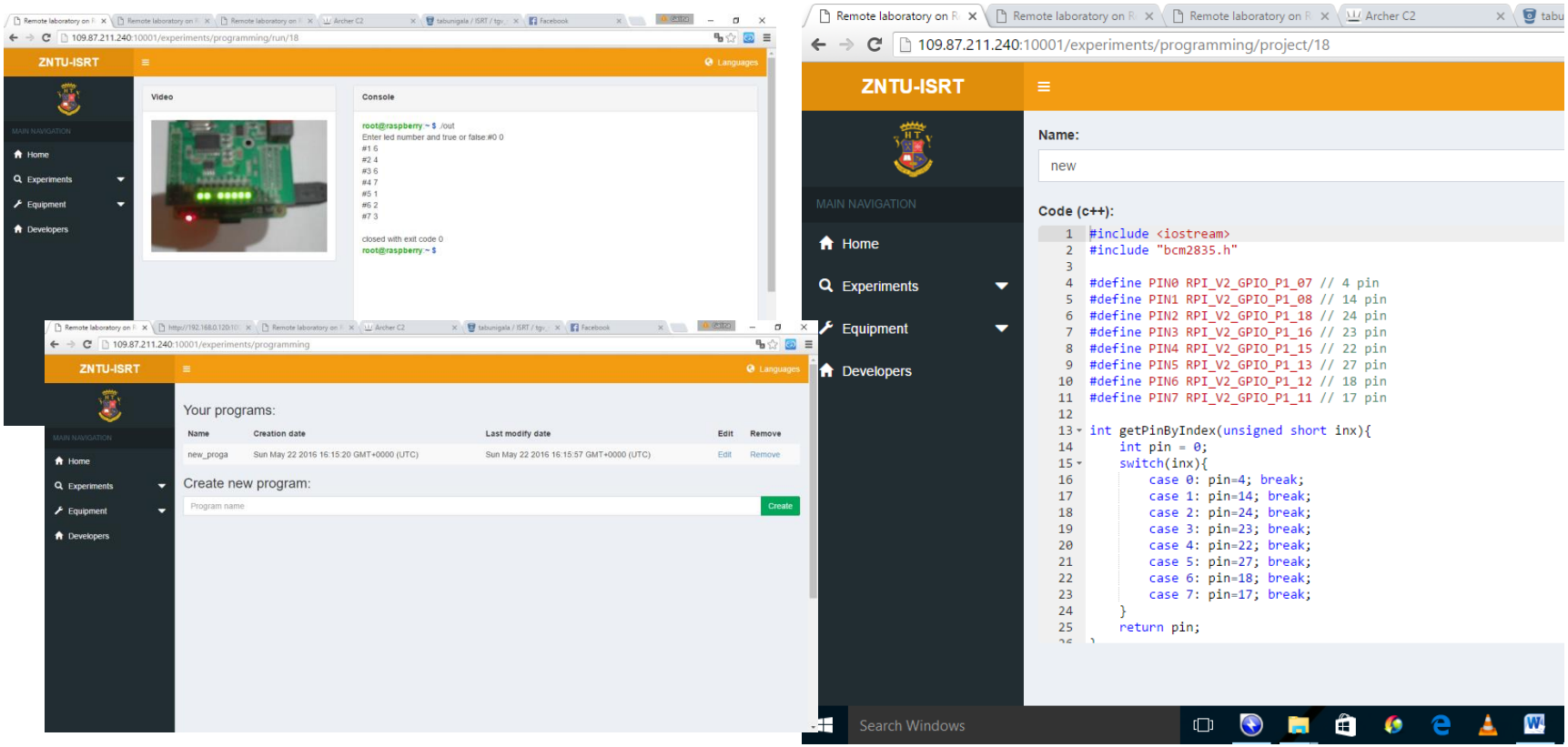
Nodejs
JavaScript





Co-funded by the
Tempus Programme
of the European Union

Programming with C++



Supported Operational Systems

- Raspbian
- OpenELEC Pidora
- Arch Linux ARM
- Kali Linux
- Windows 10

LOG with SSH client for Windows

PuTTY

192.168.1.201/209
login pi
password raspberry

Xshell
www.p30download.com

Static Network Settings

Pathname Description

/etc/network/interfaces Main configuration file for networks

/etc/wpa_supplicant/wpa_supplicant.conf Authentication information

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
        address 192.168.1.201
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
```

Web-Server on Raspberry Pi

LAMP (Linux, Apache, MySQL, PHP)

apt-get update

INSTALL APACHE apt-get install apache2

TEST THE WEB SERVER http://localhost/

default web page is just a HTML

sudo nano /var/www/html/index.html

First html file

```
<html>
<head>
        <title>Raspberry Pi web server</title>
</head>
<body>
        Hi! Its test server at Raspberry Pi and  Raspbian
</body>
</html>
```

LAMP (Linux, Apache, MySQL, PHP)

INSTALL THE PHP AND MYSQL

sudo apt-get install mysql-server

sudo apt-get install php5

sudo apt-get install php5-mysql

TEST

sudo nano index.php

<?php echo “My Raspi World !!!"; ?>

sudo rm index.html

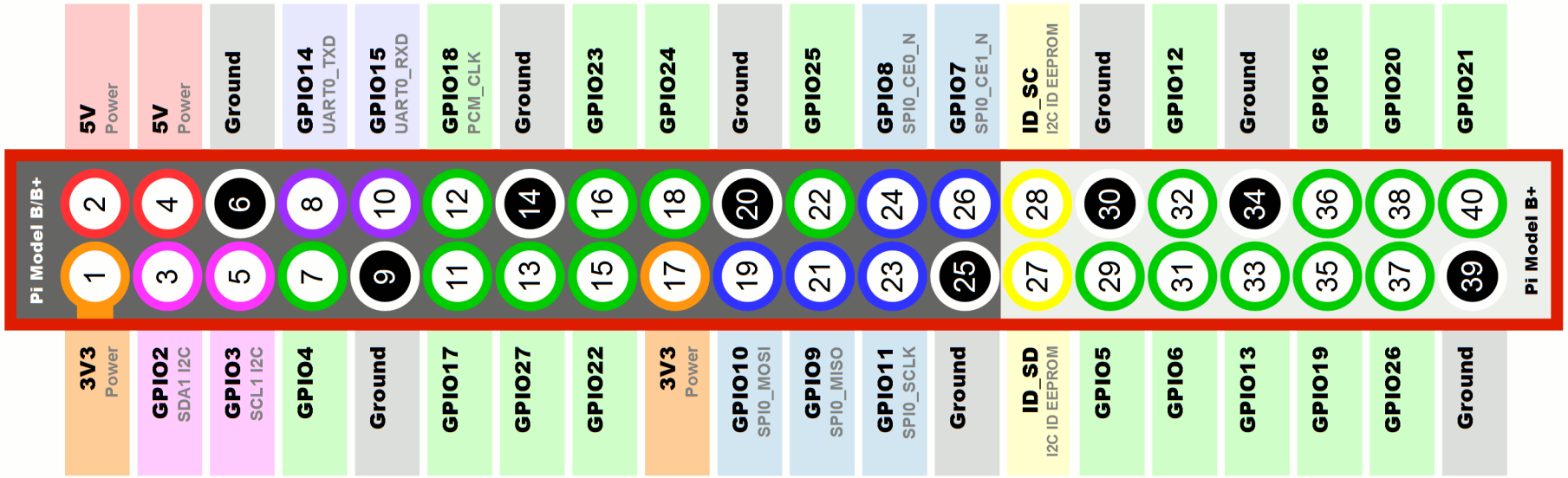
RESTART

sudo /etc/init.d/apache2 reload
sudo /etc/init.d/apache2 restart

Expansion Possibilities

Co-funded by the
Tempus Programme
of the European Union

General-purpose I/O (GPIO)



Raspberry Pi as FM Transmitter

https://github.com/markondej/fm_transmitter

Apply antenna at GPIO4

Co-funded by the
Tempus Programme
of the European Union

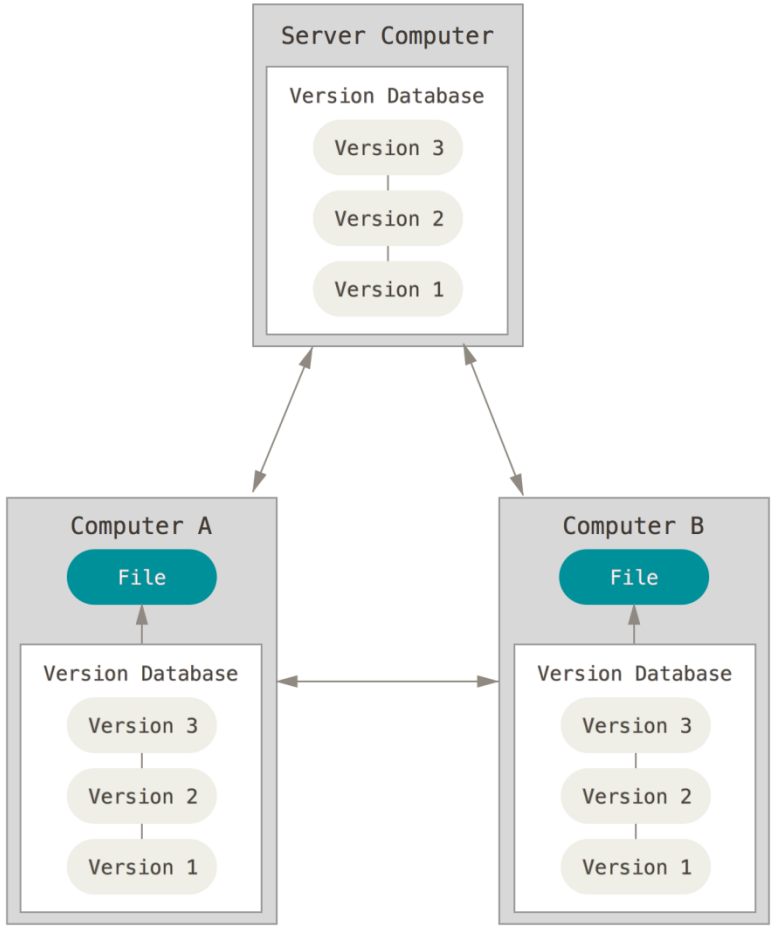
TMMA expansion board

IO
8 leds
Spi
temperaturesensor
I2c lightsensor
SPI CAN interface

github.com/bthange/Export-More.

Distributed CVS

- Git
- Mercurial
- Bazaar



Git Commands

Git Commands

git <command>

Help:

- man git < command >
- git < command > --help

git init

- git config [<file-option>]
  - --global
  - --system
  - -f config-file--file config-file
  - -l—list

Example:

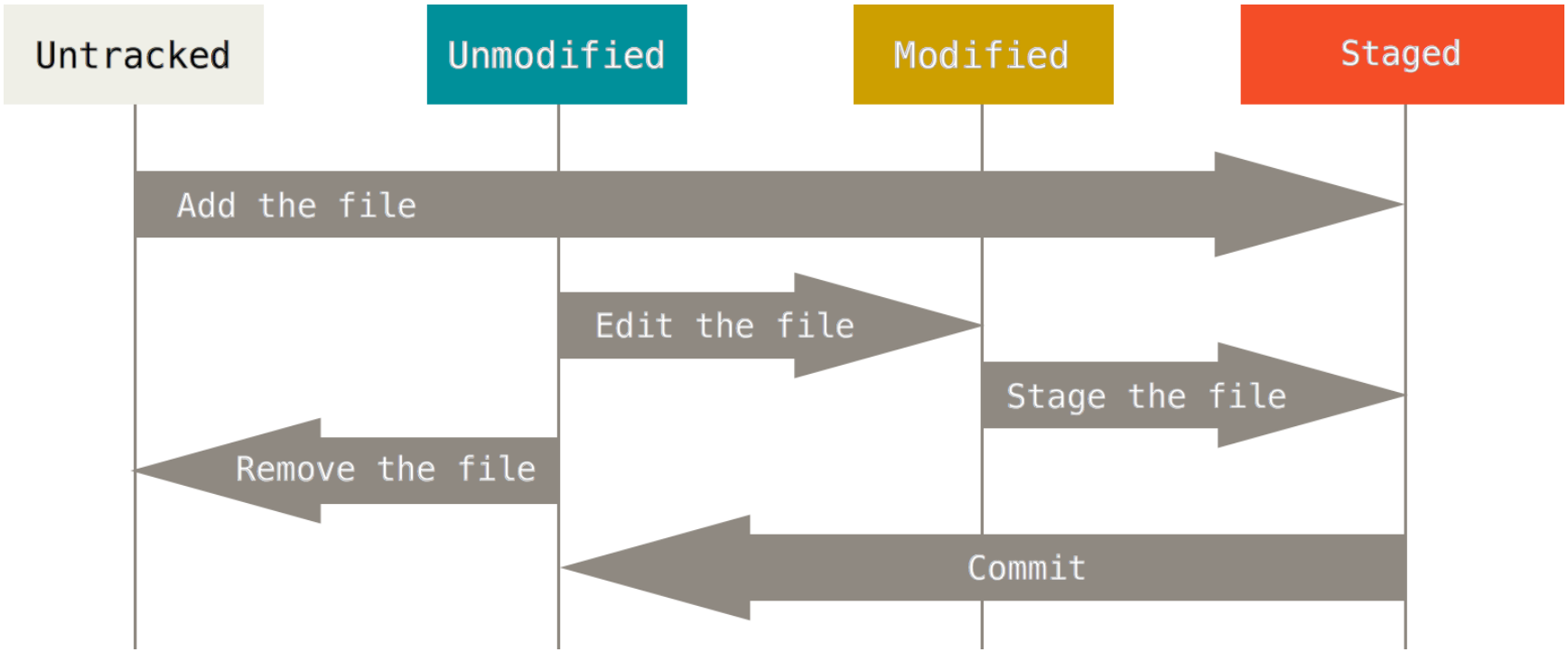
$ git init

$ git config --global user.name “John Doe“

$ git config --global user.email gohn.doe@gmail.com

$ git config --list

File LifeCycle

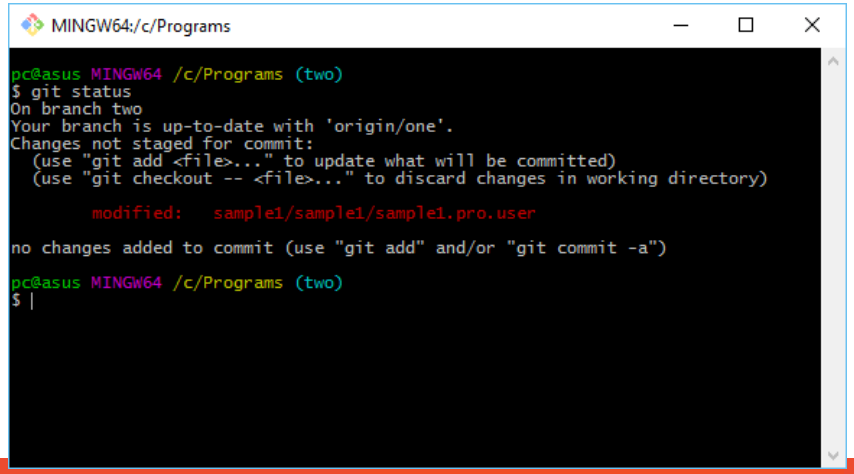


Co-funded by the
Tempus Programme
of the European Union

Git Commands

- git **add** ' <file_name> or <folder name>
- git **rm** --cached <file_name> or <имя_директории>
  - ' -f ' или ' --force ' <file name> or < folder name >
- git **status**
  - **-u**
- git **commit** <file name>
  - -m "<description>„
  - ' -a
- git commit <file_name>
- git reset

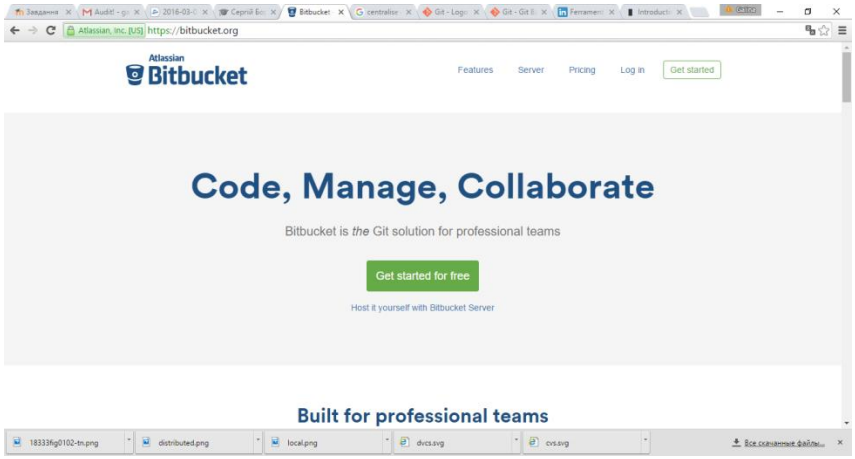
- $ git add *.*
- $ git commit -m " first commit“
- $ git status



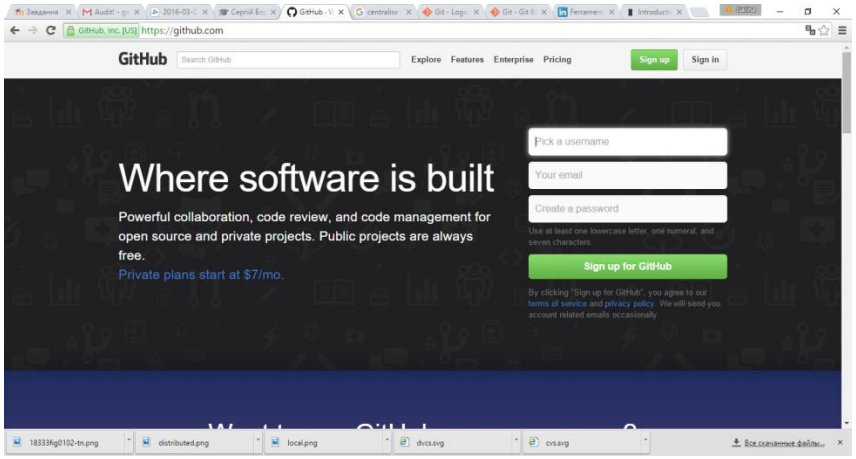
Co-funded by the
Tempus Programme
of the European Union

Remote Repositories

- BitBucket



- GitHub



Git commands for Remote Repositories

$ git remote

origin

$ git fetch [name of remote server]

$ git pull

$ git push [remote server] [branch]

$ git remote rename

$ git remote rm

git config --global user.email "my_email@mail.com"

git config --global user.name "my_nickname"

git config --global push.default simple

nothing

current

upstream

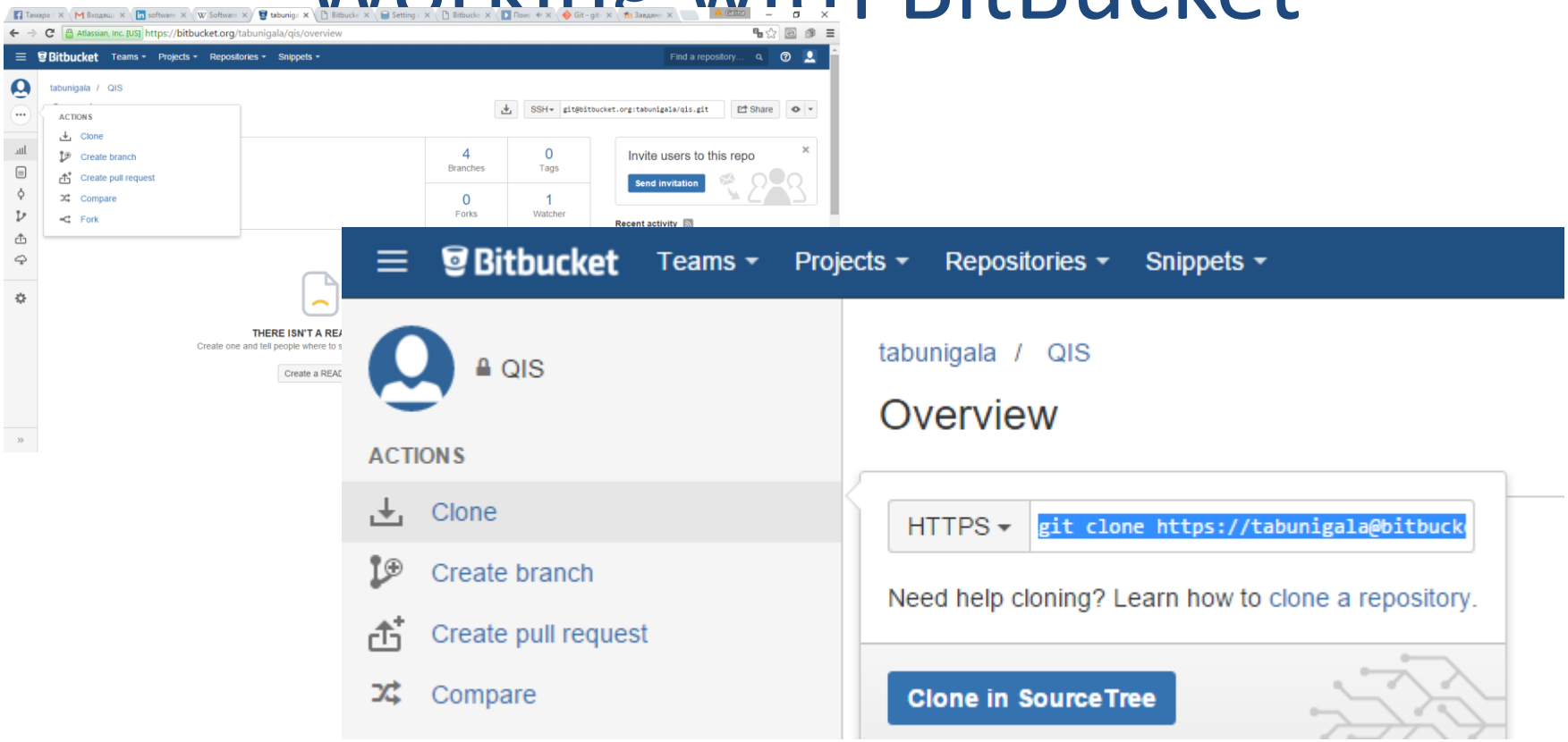
simple

matching

git clone [URL]

Co-funded by the
Tempus Programme
of the European Union

Working with BitBucket



https://tabunigala@bitbucket.org/tabunigala/mc-am.git

Co-funded by the
Tempus Programme
of the European Union

C++
Programming for Raspberry Pi

Co-funded by the
Tempus Programme
of the European Union

Manipulating TMMA expansion board

1. Library bcm2853 are provided

git clone https://tabunigala@bitbucket.org/tabunigala/mc-am.git

Password DesireForever

2. Unpack bcm2835-1.29.tar.gz

tar -zxf bcm2835-1.42.tar.gz

cd bcm2835-1.42

3. Install library

./configure

make

sudo make install

4. Unzip Embedded_OS.zip

sudo apt-get install unzip

sudo unzip Embedded_OS.zip

Thomas-More Examples

2-gpio	Leds Programming
3-lightsensor	Light Sensor
4-temperature	Temperature Sensor
9-datasheet	Broadcom bcm2835 peripherals **ADT7310 - Digital SPI Temperature Sensor** TSL256x LIGHT-TO-DIGITAL CONVERTER

Co-funded by the
Tempus Programme
of the European Union

Testing bcm2835

g++ light.c -o light -l bcm2835

sudo ./light

g++ temperature.c -o temperature -l bcm2835

sudo ./temperature

```
pi@raspberrypi ~/Export-More/3-lightsensor $ sudo ./light
13 - if 33 the device is turned on
ad value:349
pi@raspberrypi ~/Export-More/3-lightsensor $ sudo ./light
33 - if 33 the device is turned on
ad value:349
```

Led Manipulation

```
#define PIN0 RPI_V2_GPIO_P1_07  //4
#define PIN1 RPI_V2_GPIO_P1_08  //14
#define PIN2 RPI_V2_GPIO_P1_18  //24
#define PIN3 RPI_V2_GPIO_P1_16  //23
#define PIN4 RPI_V2_GPIO_P1_15  //22
#define PIN5 RPI_V2_GPIO_P1_13  //27
#define PIN6 RPI_V2_GPIO_P1_12  //18
#define PIN7 RPI_V2_GPIO_P1_11  //17
```

```
bcm2835_gpio_fsel(PIN0,
BCM2835_GPIO_FSEL_OUTP);
bcm2835_gpio_fsel(PIN0,
BCM2835_GPIO_FSEL_INPT);
```

PIN0 High

```
bcm2835_gpio_write(PIN0, HIGH);
bcm2835_gpio_set(PIN0);
```

PIN0 low

```
bcm2835_gpio_write(PIN0, LOW);
bcm2835_gpio_clr(PIN0);
```

Co-funded by the
Tempus Programme
of the European Union

Light Sensors Manipulation

TSL2561

- i2c lichtsensor

- 16-Bit Digital Output

- Low Active Power (0.75 mW Typical) with

Power Down Mode

1. Get the I2C pins in the good configuration

bcm2835_i2c_begin();

2. Change slave address

bcm2835_i2c_setSlaveAddress(0x29); // The default

3. Change baudrate

bcm2835_i2c_set_baudrate(1000); // The default

I2c write command

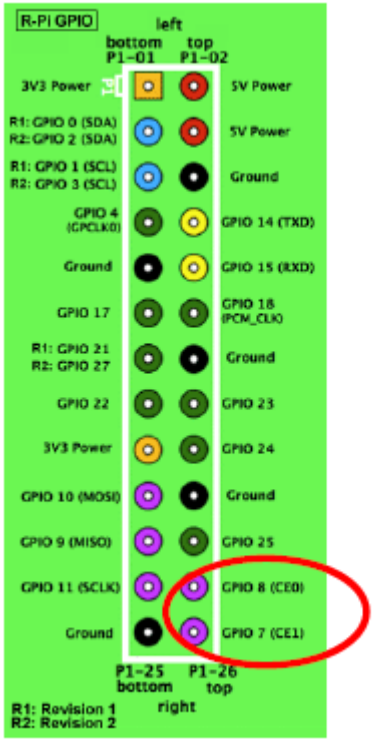
bcm2835_i2c_write(temp,1);

I2c read command

bcm2835_i2c_read(temp,1);

Temperature Sensor

- Analog ADT7310
- - Spi temperatuur sensor
- - ±0.5°C from –40°C to +105°C (2.7V to 3.6V)
- - 700 µW typical at 3.3 V in normal mode



Co-funded by the
Tempus Programme
of the European Union

Temperature Sensor Manipulation

1. Begin

```
bcm2835_spi_begin();
```

2. Configuration

```
bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST);
bcm2835_spi_setDataMode(BCM2835_SPI_MODE3);
bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_65536);
bcm2835_spi_setChipSelectPolarity(BCM2835_SPI_CS0, LOW);
```

3. Spi send and receive

```
bcm2835_spi_transfern(buffer, 2);
```

Tasks for Labs

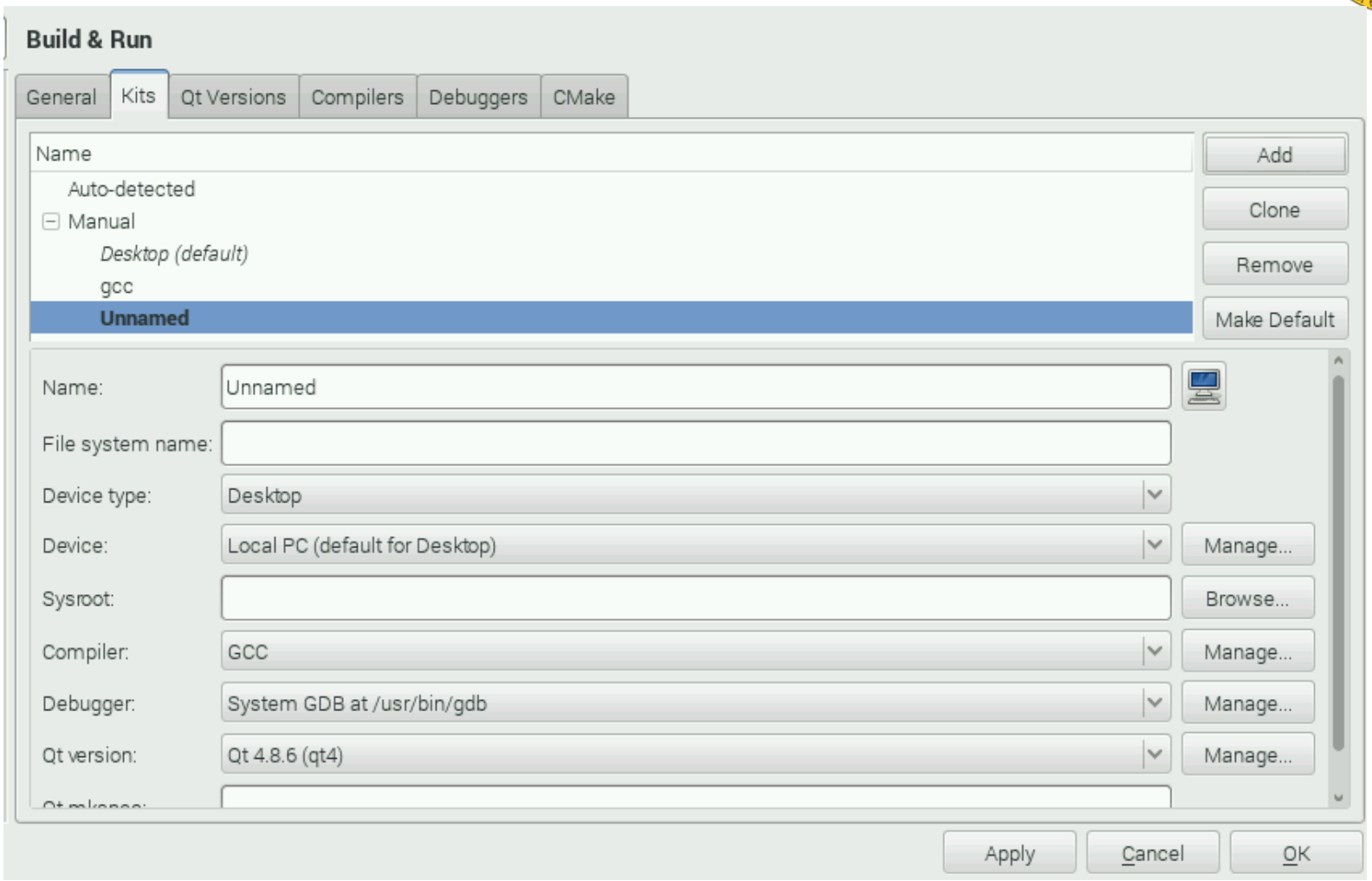
With defined delay get data from temperature sensor and show it on leds

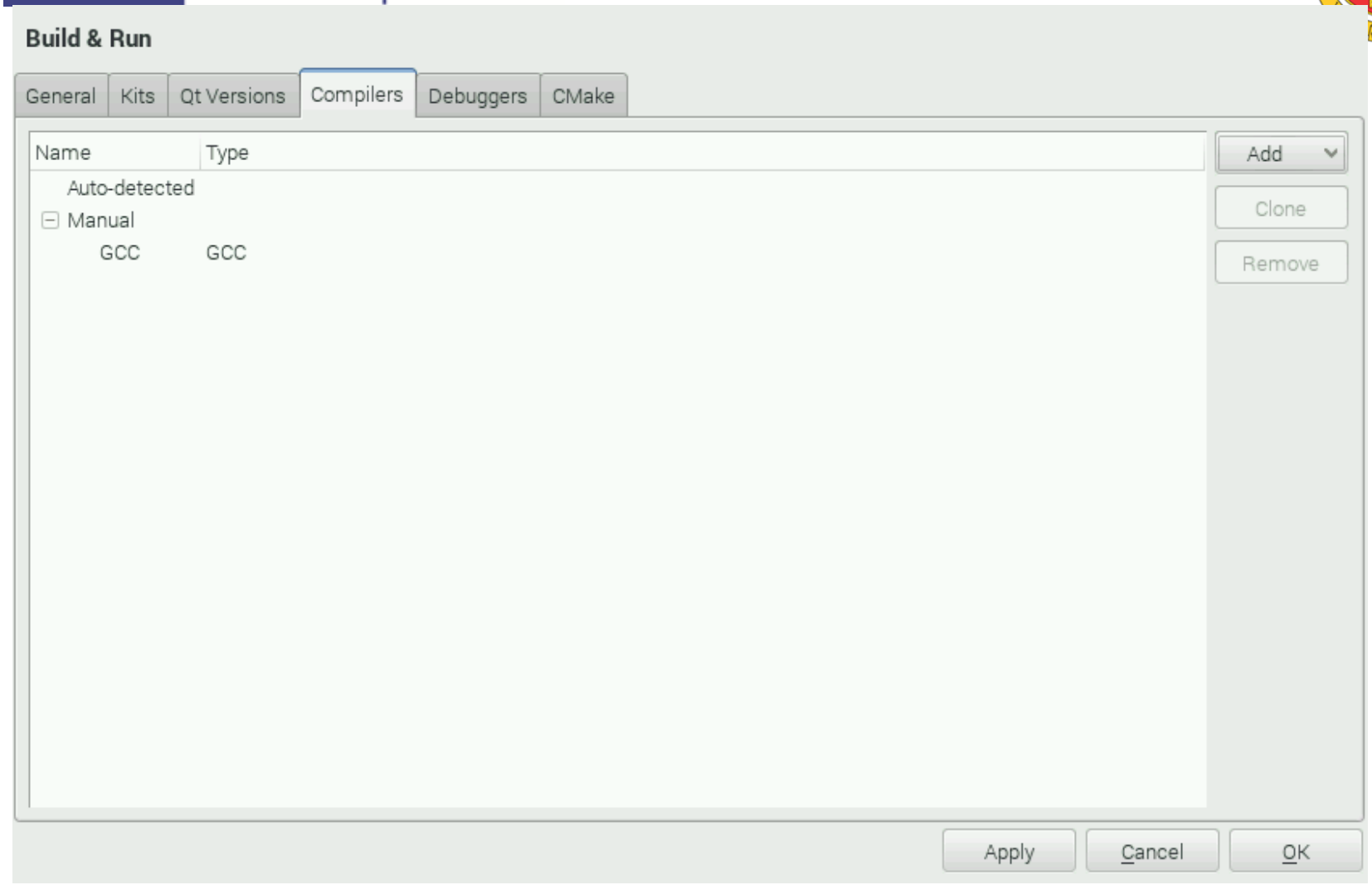
Get data from light sensor and write result in the file spec in command line

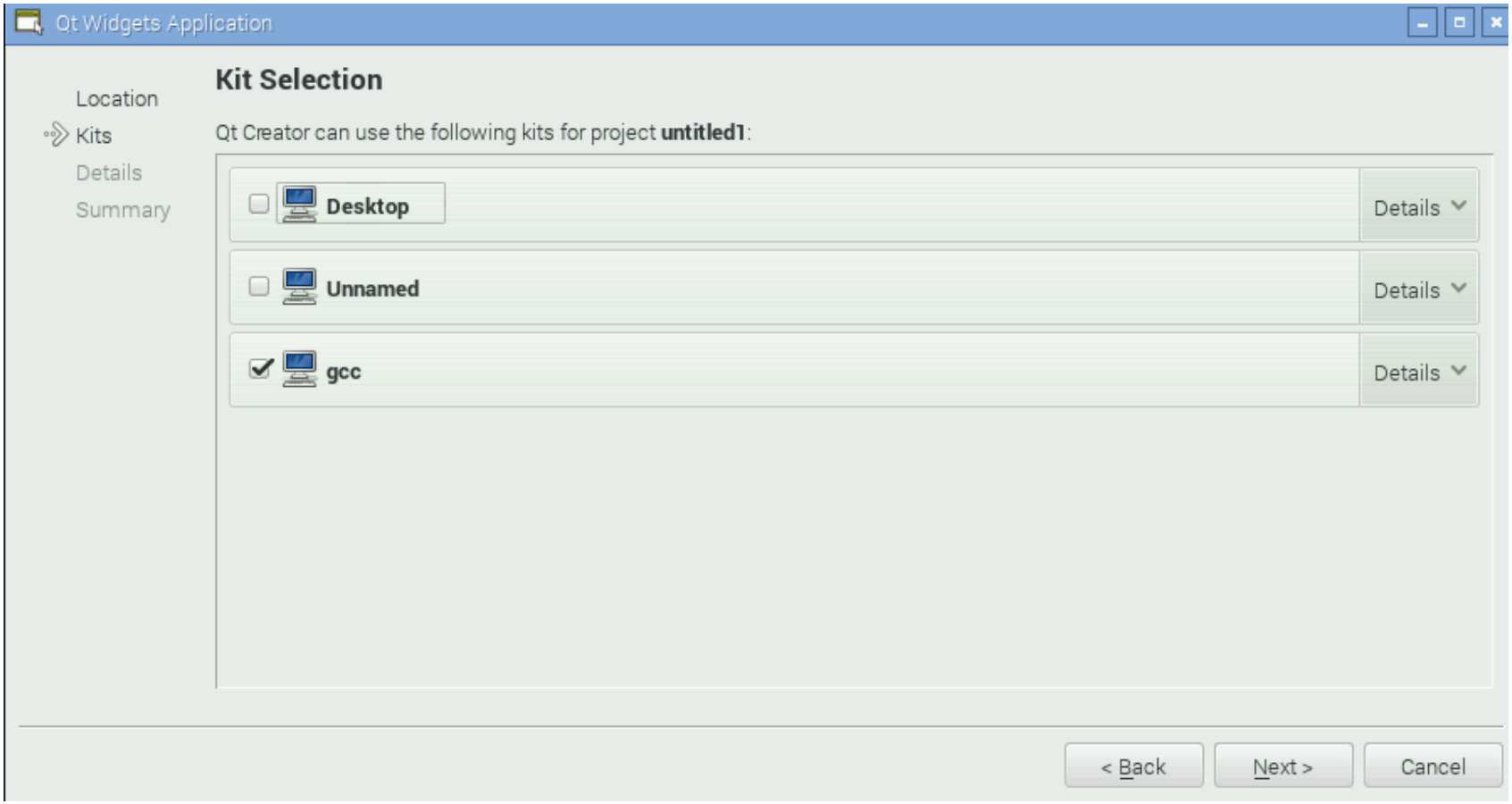
Other
Projects on Raspberry Pi

QT

- sudo apt update
- sudo apt-get install qt4-dev-tools
- sudo apt-get install qtcreator

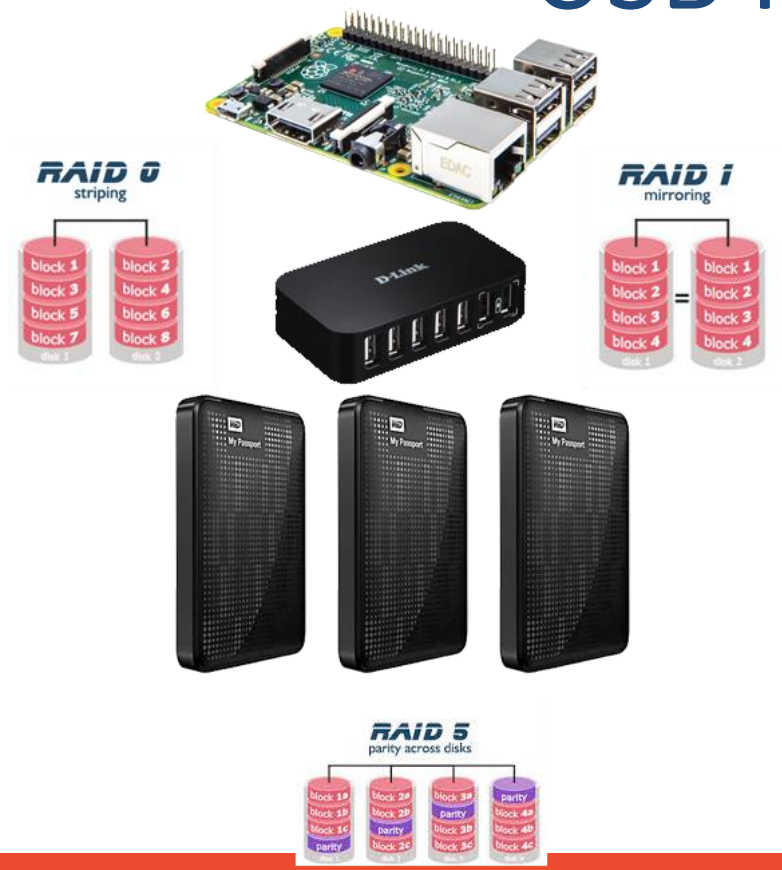

Build & Run
General
Kits
Qt Versions
Compilers
Debuggers
CMake
Name
Add
Auto-detected
Manual
Desktop (default)
gcc
Unnamed
Clone
Remove
Make Default
Name:
Unnamed
File system name:
Device type:
Desktop
Device:
Local PC (default for Desktop)
Manage...
Sysroot:
Browse...
Compiler:
GCC
Manage...
Debugger:
System GDB at /usr/bin/gdb
Manage...
Qt version:
Qt 4.8.6 (qt4)
Manage...
Apply
Cancel
OK


Build & Run
General
Kits
Qt Versions
Compilers
Debuggers
CMake
Name
Type
Auto-detected
Manual
GCC
GCC
Add
Clone
Remove
Apply
Cancel
OK


Qt Widgets Application
Location
Kits
Details
Summary
Kit Selection
Qt Creator can use the following kits for project untitled1:
Desktop
Details
Unnamed
Details
gcc
Details
< Back
Next >
Cancel

Co-funded by the
Tempus Programme
of the European Union

RASPBERRY PI RAID ARRAY WITH USB HDDS



- Connect Hard Drives to the Raspberry Pi .
- Install mdadm to create the raid assembly
- Configure the raid assembly.

RASPBERRY PI RAID ARRAY WITH USB HDDS

STEP 1 : Update the Pi

sudo apt-get update

sudo apt-get upgrade

sudo apt-get dist-upgrade

sudo reboot

STEP 2 : Connect the USB HDDs

sudo fdisk –l

STEP 3 : Install Mdadm

sudo –l

apt-get install mdadm

mdadm -Cv /dev/md0 -l0 -n2 /dev/sd[ab]1

(mdadm -Cv /dev/md0 -l1 -n2 /dev/sd[ab]1)

fdisk –l

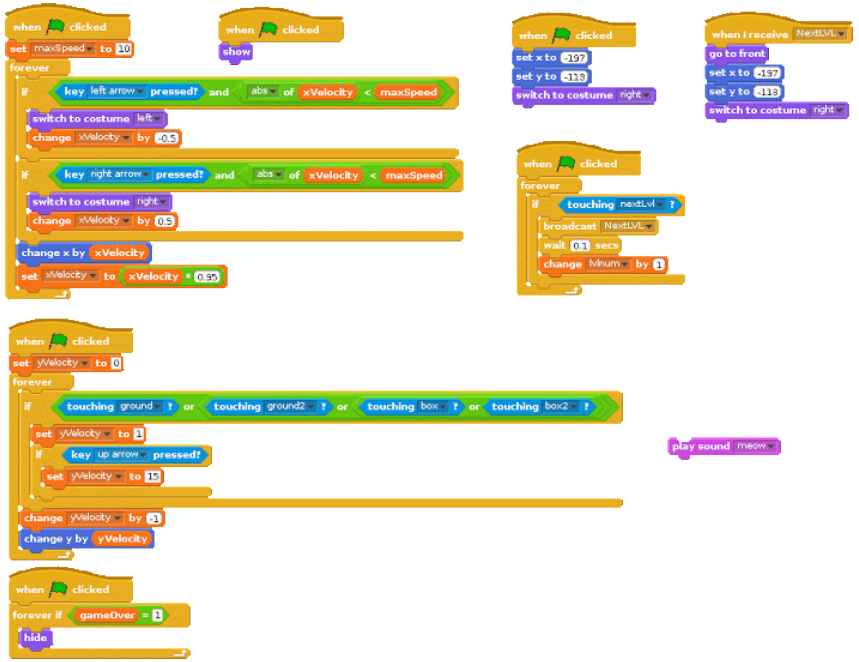
cat /proc/mdstat

mkfs /dev/md0 -t ext4

mdadm --detail /dev/md0 t

Co-funded by the
Tempus Programme
of the European Union

Scratch on Raspberry Pi



Thank You for Your Attention