



Co-funded by the
Tempus Programme
of the European Union



Manipulating with Raspberry Pi

Galyna TABUNSHCHYK

Prof. Software Tools Department

Zaporizhzhia National Technical University



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



ZAPORIZHZHYA NATIONAL TECHNICAL UNIVERSITY

Guest Lecture

Workshop

on the VHDL basics

by Ing. Dirk Van Merode MSc.

Co-funded by the Tempus Programme of the European Union

October 16, 2015
Software Department
building III, room 57
10:05 a.m.



Goal

- ✓ To explain the use of hardware description languages
- ✓ To allow the participants basic logic gates, combinational logic and sequential logic with the use of VHDL on a Xilinx FPGA-board
- ✓ To introduce the ISE software.

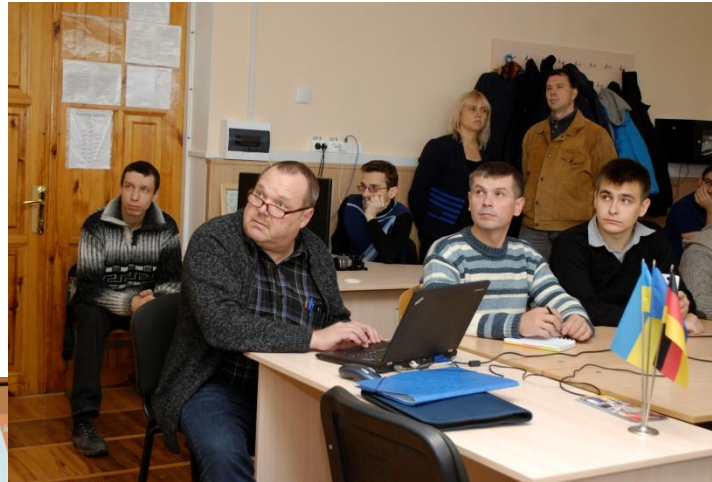
Abstract

FPGAs in embedded systems are omnipresent. They are used in a number of applications, being it ASIC-design for chip-emulation and fast time-to-market, being it in high-data-throughput telecommunication and Digital Signal processing. To work and to teach Digital System Design with FPGAs is rather complex, due to the fact that the principles behind describing hardware are somewhat different with traditional programming software. In this view, it is a good idea to start off with basic gates, to get a fundamental knowledge on the way these interesting components work.

About the Speaker

In 2002 Dirk Van Merode finished his engineering studies in Electronics to become a Master in Science. His first educational experience was in secondary education in electricity and electronics, to earn his certificate in pedagogical aptitude. Dirk moved to Leuven University College, currently renamed Thomas More University College, in 2007, to take up a teaching assignment and to do research. His field of expertise is in digital systems design, printed circuit board design and production, and audio-video production. Research topics are mainly

64, Zhukovskogo st.
Zaporizhzhya
+380(91) 769 82 6
galina.tabunshchik
www.zntu.edu.ua



ZAPORIZHZHYA NATIONAL TECHNICAL UNIVERSITY

Guest Lecture

CREO as a tool for virtual prototyping

by Dr Ing Peter Arras, PhD

Co-funded by the Tempus Programme of the European Union

October 15, 2015
Software Department
building III, room 57
10:05 a.m.



Goal

- ✓ To explain the process of integrated mechanical design and virtual prototyping
- ✓ To introduce the CREO-design software and simulating models

Abstract

Mechanical design switched from drawing oriented to model oriented design over the last decade. In a model oriented design you make a virtual prototype in the design software in which the design has all properties of the real object and behaves as the real object. This allows for virtual prototyping and testing, and is such for shorter and more robust design cycles. Less physical prototypes are necessary or can sometimes be completely eliminated, saving in time and costs.

About the Speaker

Dr Ing Peter Arras, 40 years, holds of engineering technology, graduated as engineer in electronic circuits at the University of Applied Sciences in 1985. He received his PhD degree in 2014 at the University of Duisburg-Essen in Mechanical Engineering. He worked as a research assistant in the Institute of Mechanical Engineering at the University of Duisburg-Essen. He is currently responsible for the course of mechanical design in the Institute of Mechanical Engineering at the University of Duisburg-Essen. He is also working as a consultant for mechanical design in the automotive industry and in the field of mechanical design.





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



thi

KU LEUVEN

Funded by the
Erasmus+ Programme
of the European Union



7

Armenia
23-27 May, 2016

Zaporizhzhya National Technical
University

DesIRE



Co-funded by the
Tempus Programme
of the European Union



Ministry of Education and Science of Ukraine
Certificate of Achievement

awarded to
Zaporizhzhia National Technical University

Yevgen Zadorozhnyi
Andrey Bezonov
Victoria Kosarenko
Coach: Natalia Myronova

First place

The 2015 Open All-Ukrainian Collegiate
Programming Contest

Vinnitsia, UKRAINE October 17, 2015

UCF «ASDIT»



V.V. Grabko, Prof., Dr.Sc.,
Head of the Organizing Committee, Rector of VNTU



15-18 October, 2015, VNTU, Vinnytsia





About Myself

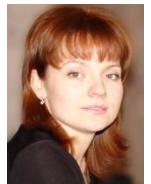
- Professor of Software Tools Department of Zaporizhzhya National Technical University, Institute of RadioElectronics and Informatics, Faculty Computer Sciences and Technologies
- supervising work of PhD students;
- **Courses:** Object Oriented Programming, Designing and Modelling of Software in Embedded 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



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



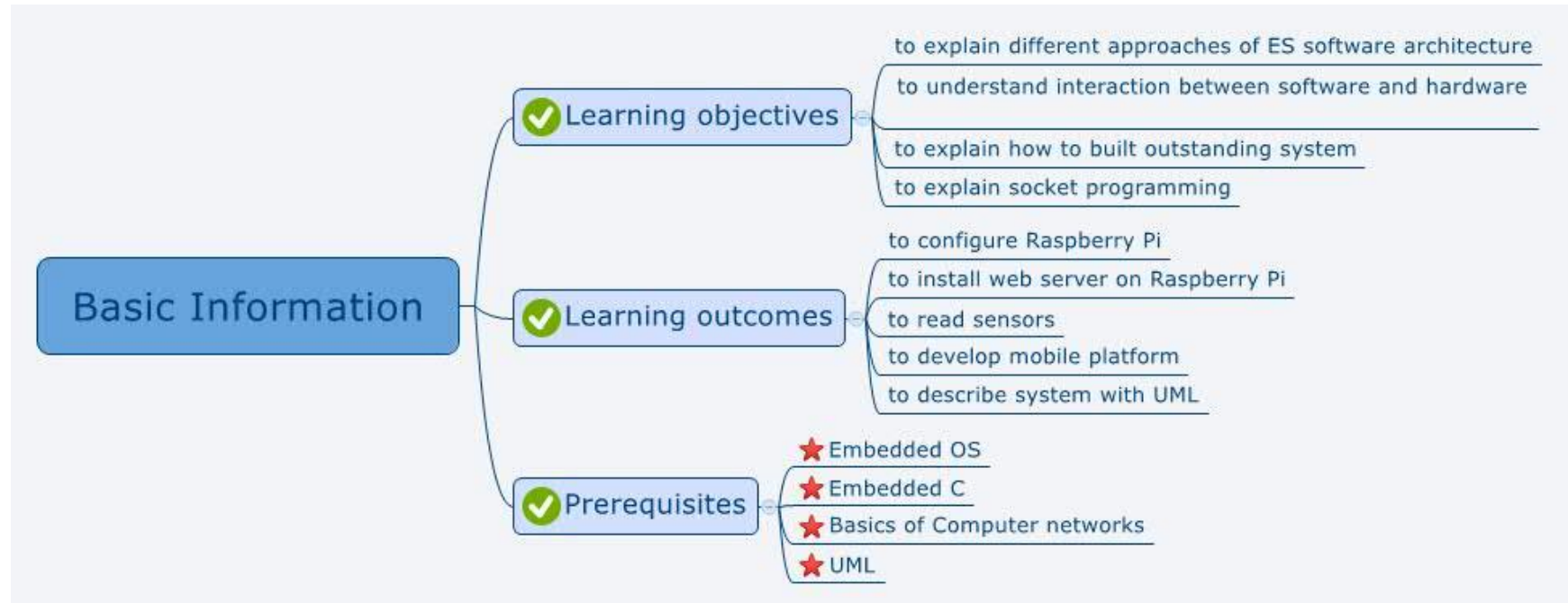
Teaching
Assistant



Natali Myronova
natali.myronova@gmail.com



Eygeniy Tverdokhlebov
junta.kristobal@gmail.com





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

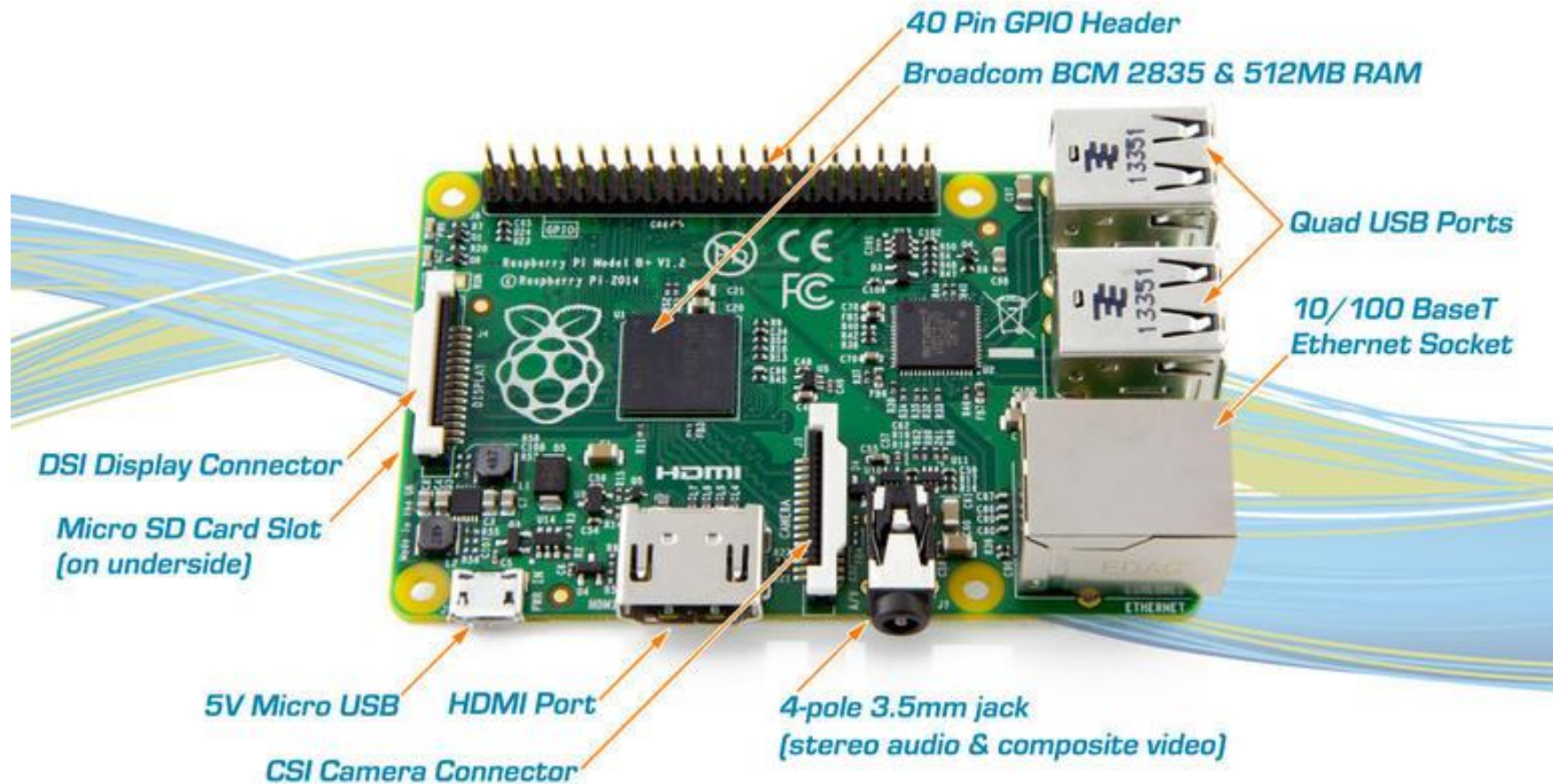
Experiments, Projects, Lab Works	Subject
Lab work 1	Configuring Raspberry Pi
Lab work 2	Installing Web-server at Raspberry Pi
Lab work 3	Developing QT application at Raspberry Pi
Lab work 4	Reading sensors from extension board
Lab work 5	Developing Project on Raspberry Pi



Co-funded by the
Tempus Programme
of the European Union



What is Raspberry Pi???





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			



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

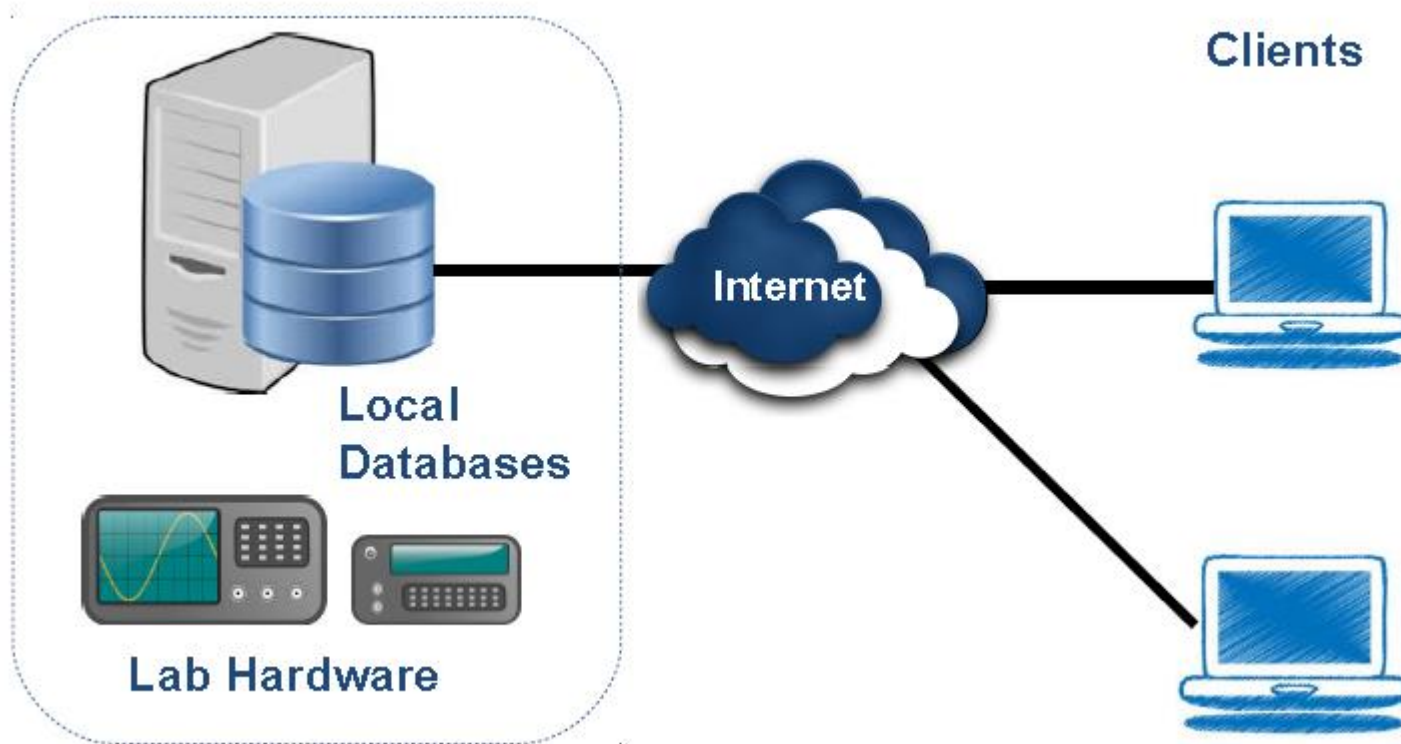
- Learn Programming
 - Scratch
 - C++
 - Python

- Web Server
- Media Server
- Cluster





REMOTE LABORATORIES





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.



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



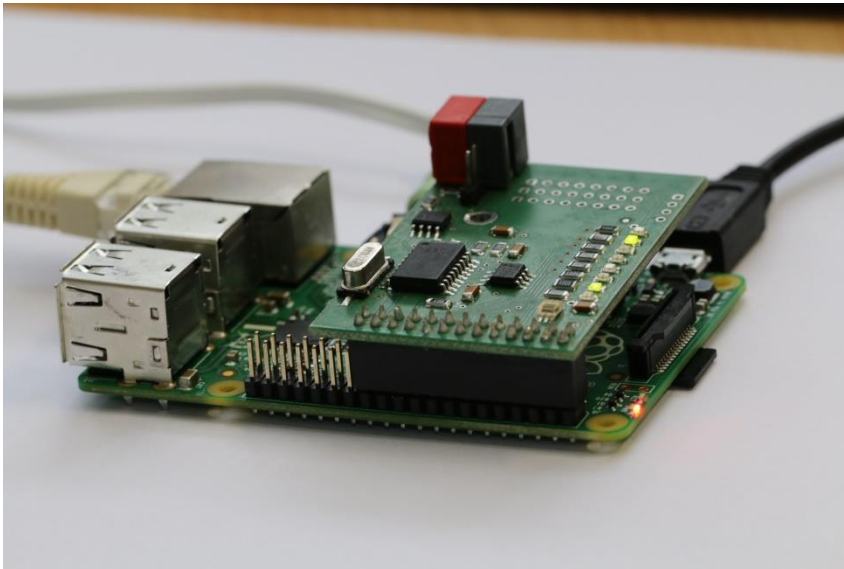
Prerequisites for students

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





New remote experiments



Pilot usage:

Master course:

Embedded Software Development

Bachelors course:

Design of Informational System

Hardware:

Raspberry Pi Model B

Expansion board

**Wifi, BLE4 adapters,
webcam**

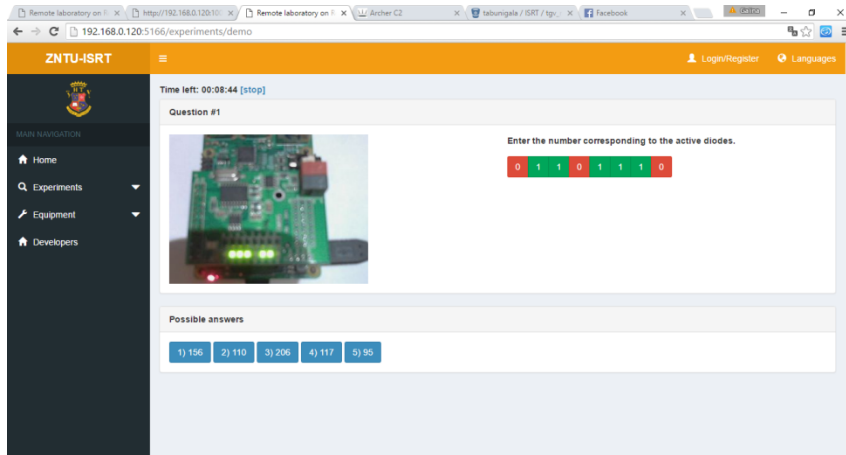
Software:

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





Two demo Modes



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

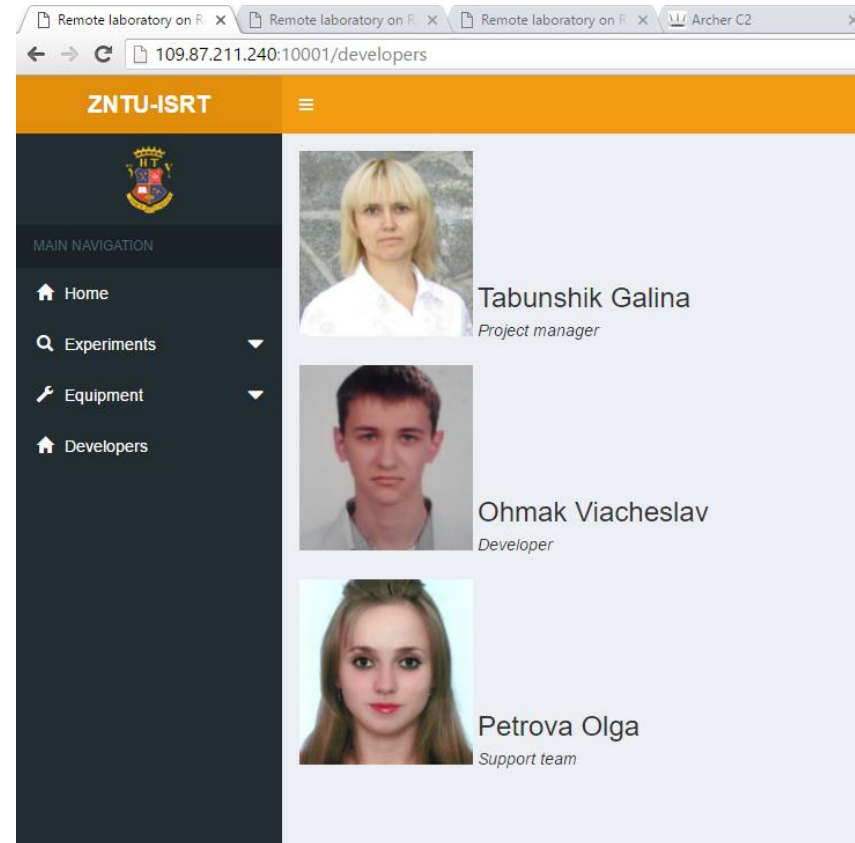
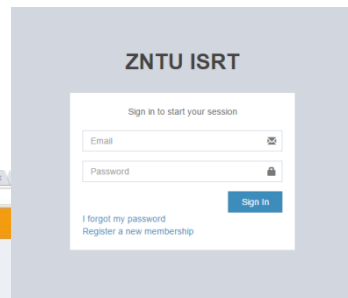
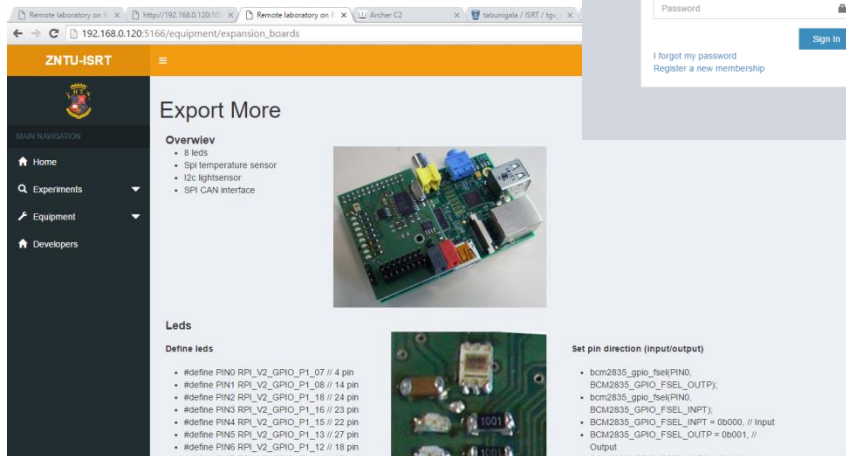


Co-funded by the
Tempus Programme
of the European Union



Web-server

Nodejs JavaScript





Programming with C++

Remote laboratory on R x Remote laboratory on R x Remote laboratory on R x Archer C2 x tabungata / ISRT / tpr. x Facebook x


109.87.211.240:10001/experiments/programming/run/18

ZNTU-ISRT

MAIN NAVIGATION

- Home
- Experiments
- Equipment
- Developers

Video



Console

```
root@raspberrypi ~$ ./out
Enter led number and true or false #0 0
#1 5
#2 4
#3 6
#4 7
#5 1
#6 2
#7 3
closed with exit code 0
root@raspberrypi ~$
```

109.87.211.240:10001/experiments/programming

ZNTU-ISRT

MAIN NAVIGATION

- Home
- Experiments
- Equipment
- Developers

Your programs:

Name	Creation date	Last modify date	Edit	Remove
new_prog	Sun May 22 2016 16:15:20 GMT+0000 (UTC)	Sun May 22 2016 16:15:57 GMT+0000 (UTC)	Edit	Remove

Create new program:

Program name

Remote laboratory on R x Remote laboratory on R x Remote laboratory on R x Archer C2 x tabungata / ISRT / tpr. x Facebook x

109.87.211.240:10001/experiments/programming/project/18

ZNTU-ISRT

MAIN NAVIGATION

- Home
- Experiments
- Equipment
- Developers

Name:

Code (c++):

```
1 #include <iostream>
2 #include "bcm2835.h"
3
4 #define PIN0 RPI_V2_GPIO_P1_07 // 4 pin
5 #define PIN1 RPI_V2_GPIO_P1_08 // 14 pin
6 #define PIN2 RPI_V2_GPIO_P1_18 // 24 pin
7 #define PIN3 RPI_V2_GPIO_P1_16 // 23 pin
8 #define PIN4 RPI_V2_GPIO_P1_15 // 22 pin
9 #define PIN5 RPI_V2_GPIO_P1_13 // 27 pin
10 #define PIN6 RPI_V2_GPIO_P1_12 // 18 pin
11 #define PIN7 RPI_V2_GPIO_P1_11 // 17 pin
12
13 int getPinByIndex(unsigned short inx){
14     int pin = 0;
15     switch(inx){
16         case 0: pin=4; break;
17         case 1: pin=14; break;
18         case 2: pin=24; break;
19         case 3: pin=23; break;
20         case 4: pin=22; break;
21         case 5: pin=27; break;
22         case 6: pin=18; break;
23         case 7: pin=17; break;
24     }
25     return pin;
26 }
```



Supported Operational Systems

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



Co-funded by the
Tempus Programme
of the European Union



LOG with SSH client for Windows



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_sup
plicant.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
```



Co-funded by the
Tempus Programme
of the European Union



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
```



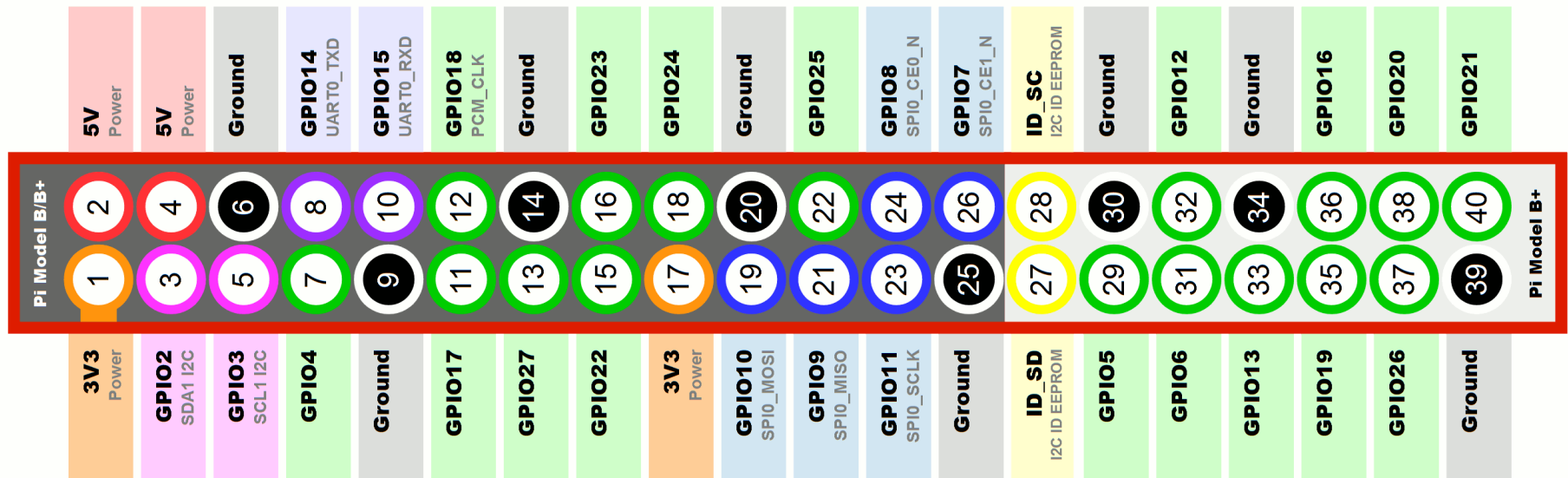
Co-funded by the
Tempus Programme
of the European Union



Expansion Possibilities



General-purpose I/O (GPIO)





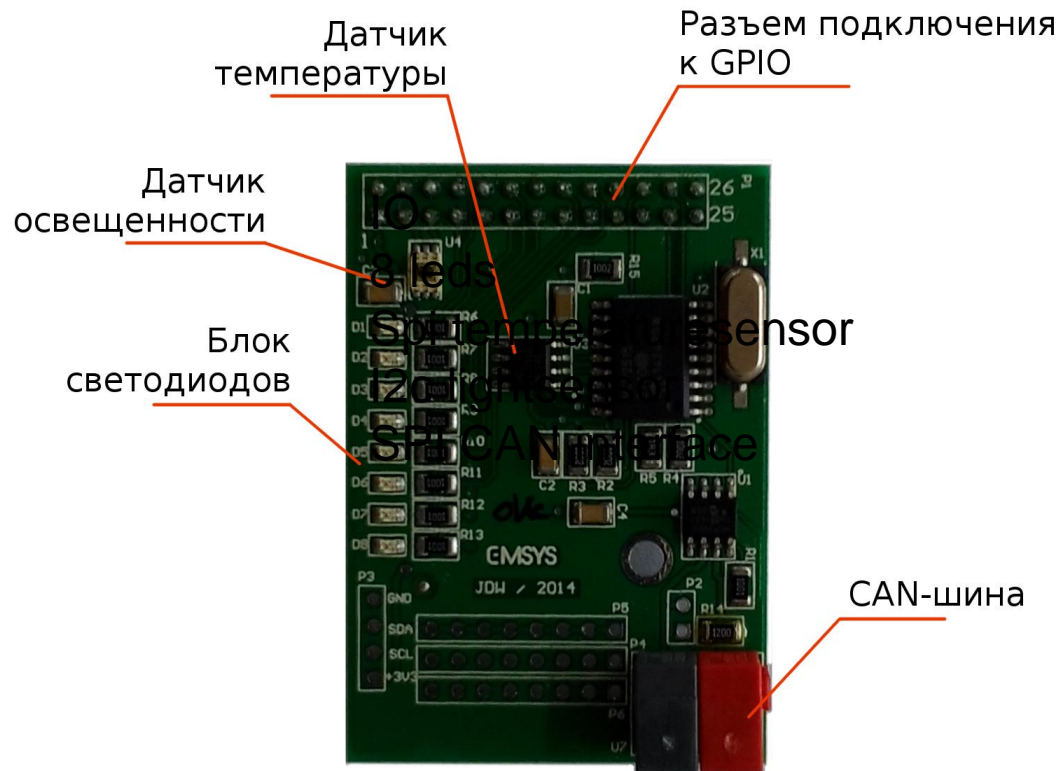
Raspberry Pi as FM Transmitter

https://github.com/markondej/fm_transmitter

Apply antenna at GPIO4



TMMA expansion board



IO
8 leds
Spi
temperaturesensor
I2c lightsensor
SPI CAN interface

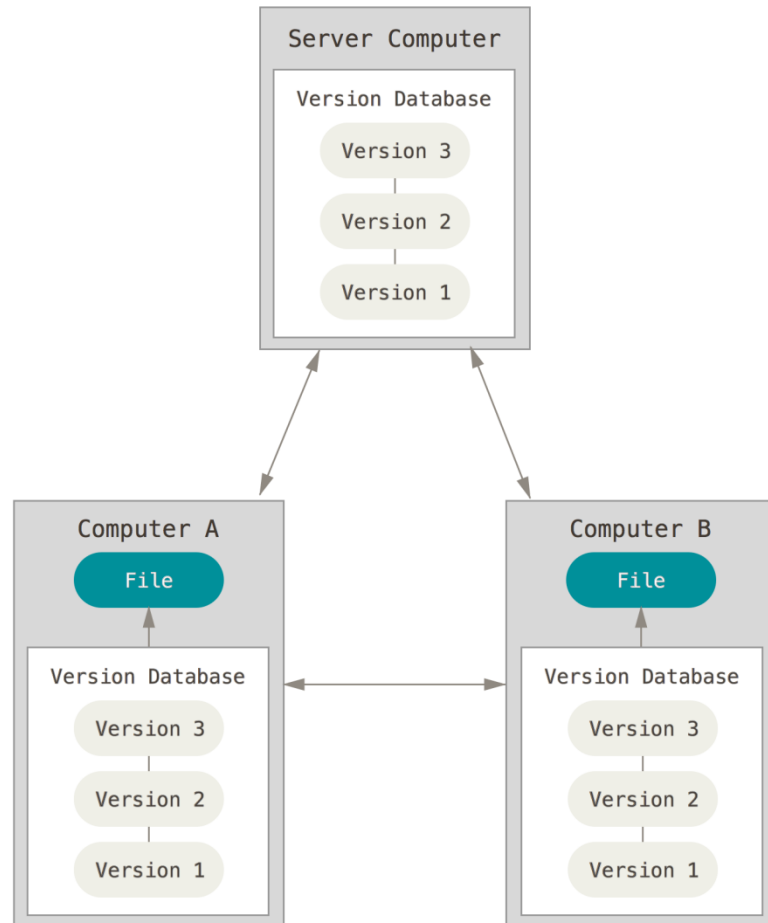
github.com/bthange/Export-More.



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
```

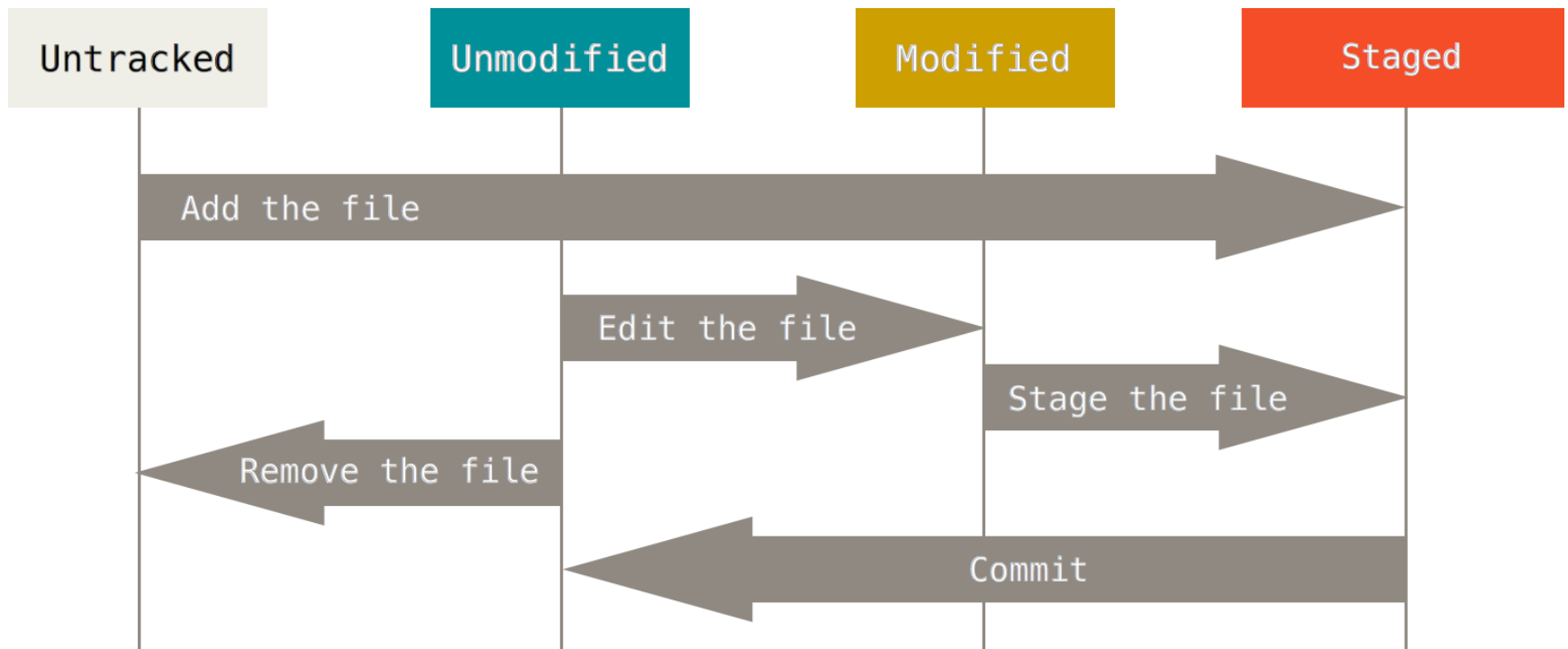
```
$ git config --global user.name "John Doe"
```

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

```
$ git config --list
```



File LifeCycle





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`

```
MINGW64/c/Programs
pc@asus MINGW64 /c/Programs (two)
$ git status
On branch two
Your branch is up-to-date with 'origin/one'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   sample1/sample1/sample1.pro.user

no changes added to commit (use "git add" and/or "git commit -a")
pc@asus MINGW64 /c/Programs (two)
$ |
```

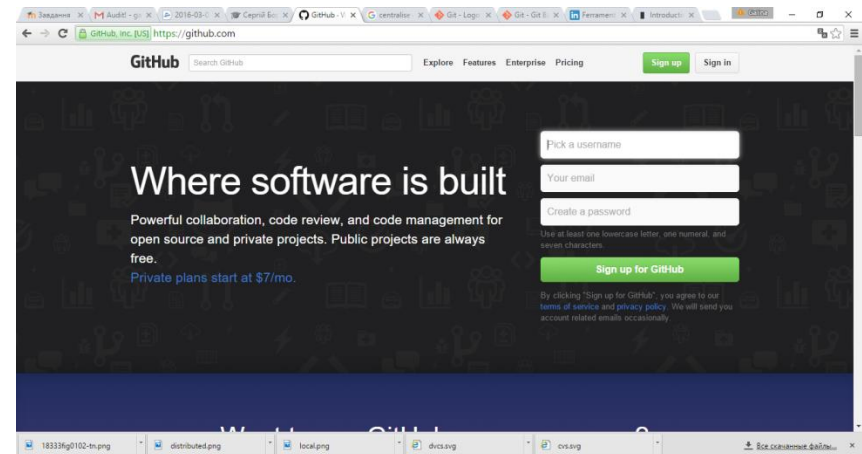
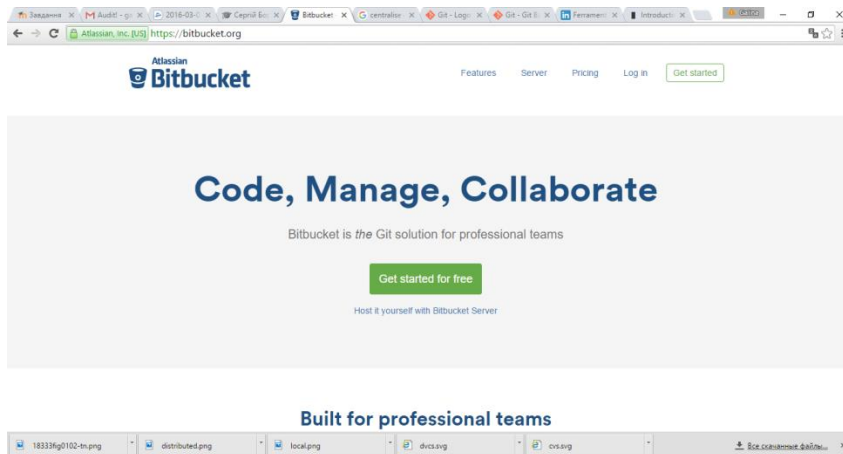


Co-funded by the
Tempus Programme
of the European Union



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

The screenshot displays the BitBucket web interface for a repository named 'tabunigala / QIS'. The interface includes a sidebar with navigation links, a main content area with repository statistics (4 branches, 0 tags, 0 forks, 1 watcher), and a 'Clone' button. A modal window is open showing the clone command: `git clone https://tabunigala@bitbucket.org:tabunigala/qis.git`.

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



Co-funded by the
Tempus Programme
of the European Union



C++ Programming for Raspberry Pi



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



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);
```




Light Sensors Manipulation

TSL2561

- i2c lightsensor
- 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(10  
00); // The default
```

I2c write command

```
bcm2835_i2c_write(temp,1);
```

I2c read command

```
bcm2835_i2c_read(temp,1);
```




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_transfer(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



Co-funded by the
Tempus Programme
of the European Union



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

Auto-detected

☐ Manual

Desktop (default)

gcc

Unnamed

Add

Clone

Remove

Make Default

Name: Unnamed

File system name:

Device type: Desktop

Device: Local PC (default for Desktop)

Sysroot:

Compiler: GCC

Debugger: System GDB at /usr/bin/gdb

Qt version: Qt 4.8.6 (qt4)

Qt plugins:

Manage...

Browse...

Manage...

Manage...

Manage...

Apply Cancel OK



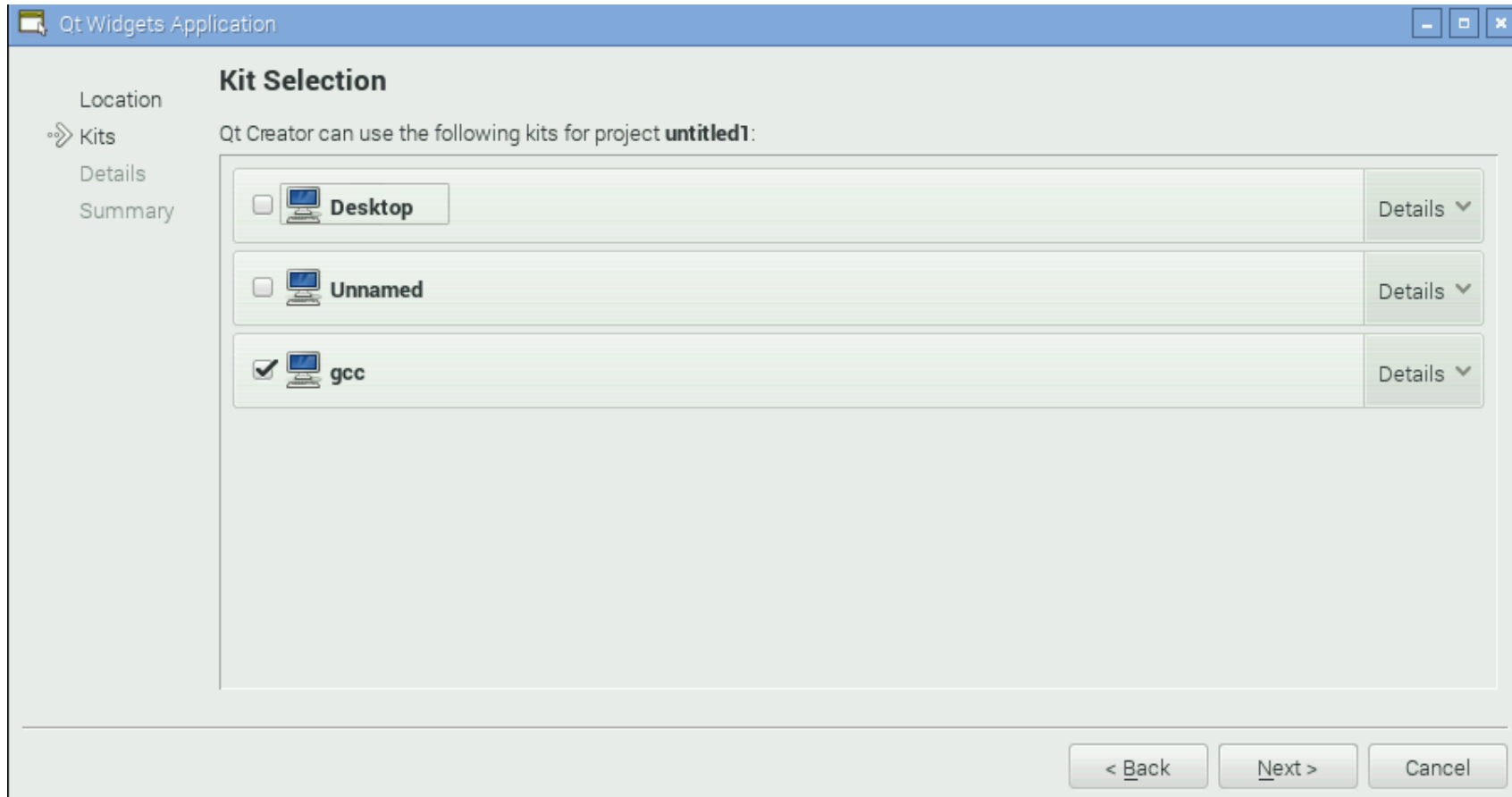
Build & Run

General Kits Qt Versions **Compilers** Debuggers CMake

Name	Type
Auto-detected	
<input type="checkbox"/> Manual	
GCC	GCC

Add Clone Remove

Apply Cancel OK



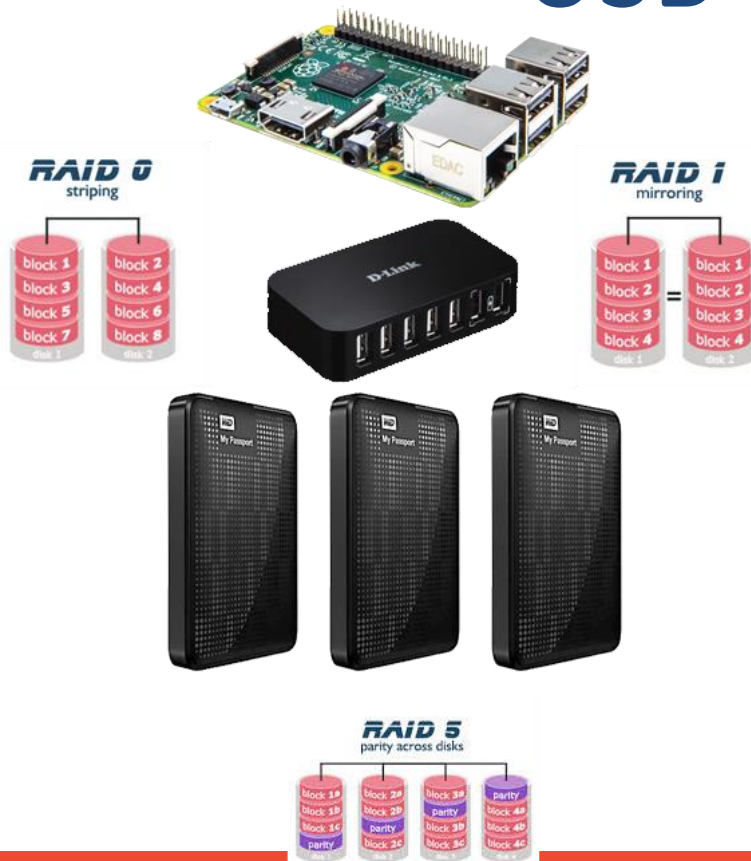


Co-funded by the
Tempus Programme
of the European Union



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
```

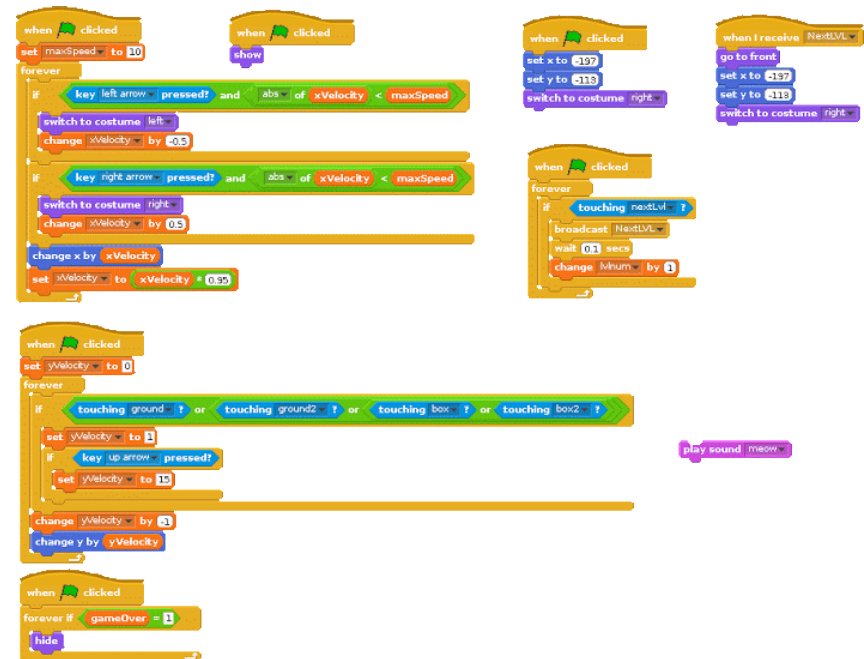
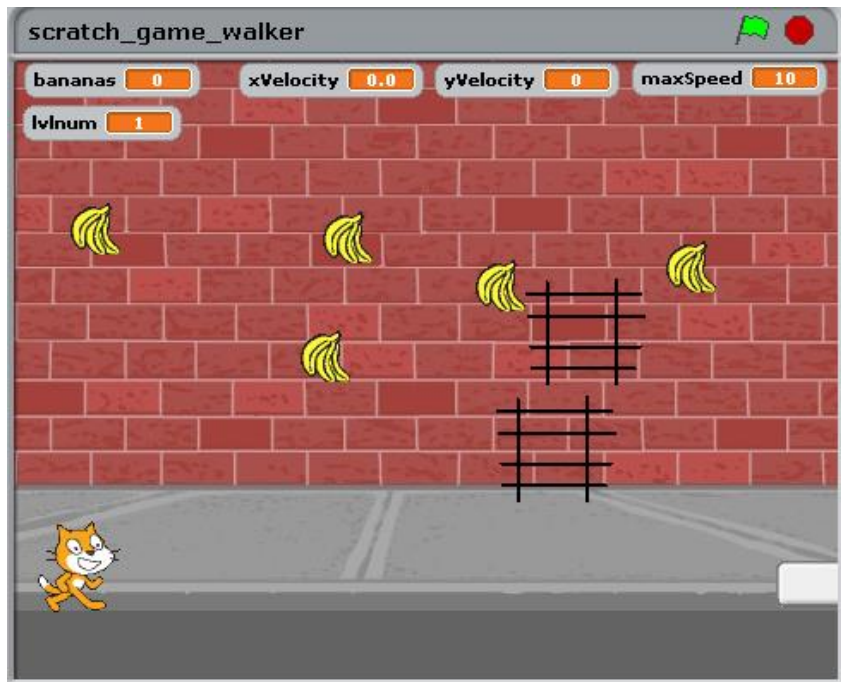
STEP 3 : Install Mdadm

```
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
```



Scratch on Raspberry Pi





Co-funded by the
Tempus Programme
of the European Union



Thank You for Your Attention