A course on embedded systems at the University of Cape Coast, Ghana

Uli Raich
Formally CERN, Geneva
now retired

African School of Fundamental Physics and Applications

Long time ago ...

- Studied Mathematics and Physics at the University of Karlsruhe (Germany) to become a high school teacher
- Came to CERN as a doctoral student
- After getting my PhD from the University of Karlsruhe I stayed at CERN and worked in the accelerator field (Linear accelerators, accelerator control, beam diagnostics)
- Took part in a technical training course on micro processors, a totally new and exciting technology
- Was asked to participate in a course on microprocessors for physicists and engineers of the developing world at the Abdus Salam ICTP, Trieste as an instructor
- Prof. Salam himself initiated the course
- The course was so successful that it was pursued and continually updated during more than 30 years

Change in Computer over a lifetime

- When I was a undergraduate student I used the university mainframe computer occupying a big room and costing several 100 kUS$
- As a doctoral student we had a minicomputer costing ~ 80 kUS $

Specifications:

- 16 bit CPU
- 128 kBytes of RAM
- 600 Mbytes of hard disk
- 1 serial terminal
- Camac interface to control the entire linear accelerator

Microprocessors

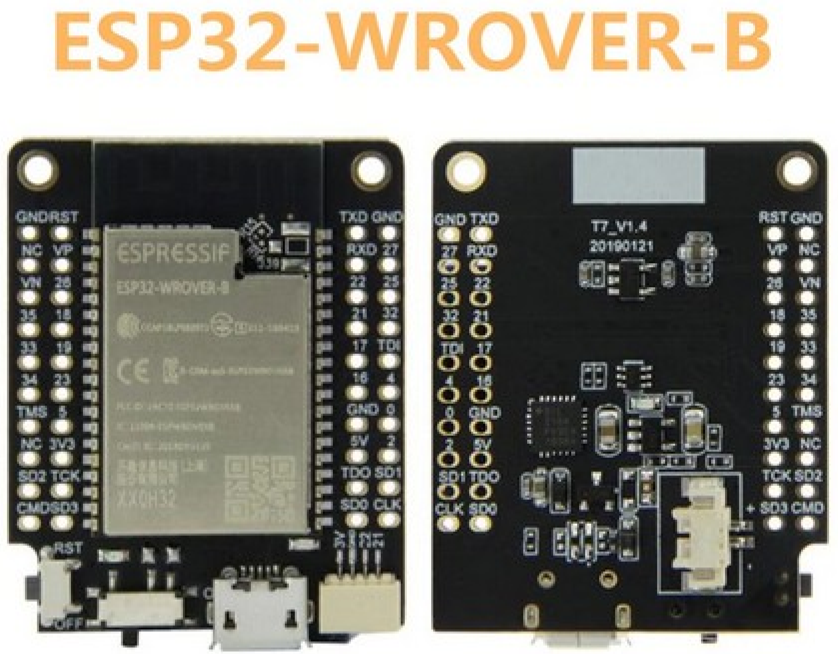
- When I started to work on microprocessors in the early 1980’s these were very simple devices containing just a single 8 bit CPU (1 MHz) and address and data lines + a few control lines to interface to external memory and I/O interfaces, typically : a serial and a parallel port.

Motorola MC6800L

- The price for the chip was 176 US $.
- The development board was sold for 300 US $. You got 256 bytes of RAM, 2kBytes of EPROM and the serial and parallel port.
- Programming was done in assembly language or even straight machine code.
- **No Internet!**

The micro-controller I use today

- Dual core 32 bit CPU (80 – 240 MHz)
- 520 kB SRAM + 8 MB PSRAM
- 4MB flash
- Interfaces:
  I2C, I2S, SPI, GPIO, ADC, DAC
- Can run on battery
- WiFi: 802.11 b/g/n
- BlueTooth and BlueTooth Low Energy
- Cost: 5.52 Euros



Students became lecturers

TECHSTORM 2018 ALL INCLUSIVE
Wikimedia is looking for new coders!
Would you like to help improve our software as a volunteer? **Read more!**

10.

Nii Quaynor

From Wikipedia, the free encyclopedia

For the Ghanaian rapper whose full name is Nii Addo Quaynor, see Tinny (musician).

Prof. Nii Narku Quaynor is a scientist and engineer who has played an important role in the introduction and development of the Internet throughout Africa.

Nii Narku Quaynor	
Born	Nii Narku Quaynor
Occupation	Computer Scientist
Known for	Developing telecommunications and Internet in Africa
Title	Professor


Prof. Nii Narku Quaynor

Contents [hide]

Biography [edit]

Prof. Quaynor graduated in engineering science from Dartmouth College in 1972 and received a Bachelor of Engineering degree from the Thayer School of Engineering there in 1973. He then studied Computer Science, obtaining an M.S. from the State University of New York at StonyBrook in 1974 and a Ph.D. from the same institution in 1977. He attended Kinbu, Adisadel College(citation needed) and Achimota School in Ghana

He is one of the founding members of the Computer Science Department at the University of Cape Coast in Ghana, and continues to hold a professorship there.[1] He is also a member of the Council of the University of

An offer

I had the idea to terminate my professional career with a stay at a university in the developing world and give back a bit of what I had received during the 30 years of microprocessor courses.

Then Nii asked me:

- Can you come to Cape Coast for 3 weeks to help the university set up a course on embedded systems?
- … or may be 6 weeks?
- Well, best would be you come for a full semester of 4 months!

The Deal

I provide:

- Setup of the embedded systems lab with 15 experimental stations
- 2h of formal lectures per week for 12 weeks
- 4h of practical sessions per week

The university

- Provides the course material (lab hardware and Internet infrastructure)
- Pays the flight and living expenses
- Provides housing
- Frees at least one local lecturer to work with me on the course. This lecturer takes over the lab and further development of the course after my departure

The goal

- Have a working lab for 15 students with up to date development software on PC computers
- 12 experiments with exercise sheets and working solutions
- Documentation of the whole course on the WEB with the possibility of easy extension
  - Lecture slides
  - Exercise sheets and solutions that can be downloaded and executed on the hardware
  - Description of sensors and actuators including data sheets for interface chips and circuit diagrams
- Final course examination
- Transmission of knowledge to the local lecturers

Hard- and Software selection

On the hardware side we had 2 contenders:

- Arduino
- Raspberry Pi

Arduino advantages

The Arduino is essentially a standardized software development system (IDE, Integrated Development Environment) running on any PC or Mac. It uses its own dialect of C++ and works on a number of hardware platforms:

- Atmel AVR processors
- ST: STM32 ARM processors
- Espressif ESP8266 and ESP32 processors

Advantages:

- Very popular among universities and the hobbyists
- Very cheap hardware (typically less than 10 US $ for the CPU card)
- Drivers for almost any sensor or actuator are available
- Uses plug-on shields

Arduino disadvantages

- Can only be programmed in its specific C++ dialect
- Only cross compilation is possible
- Internet access needs additional hardware (but is possible)
- No hard disk or equivalent

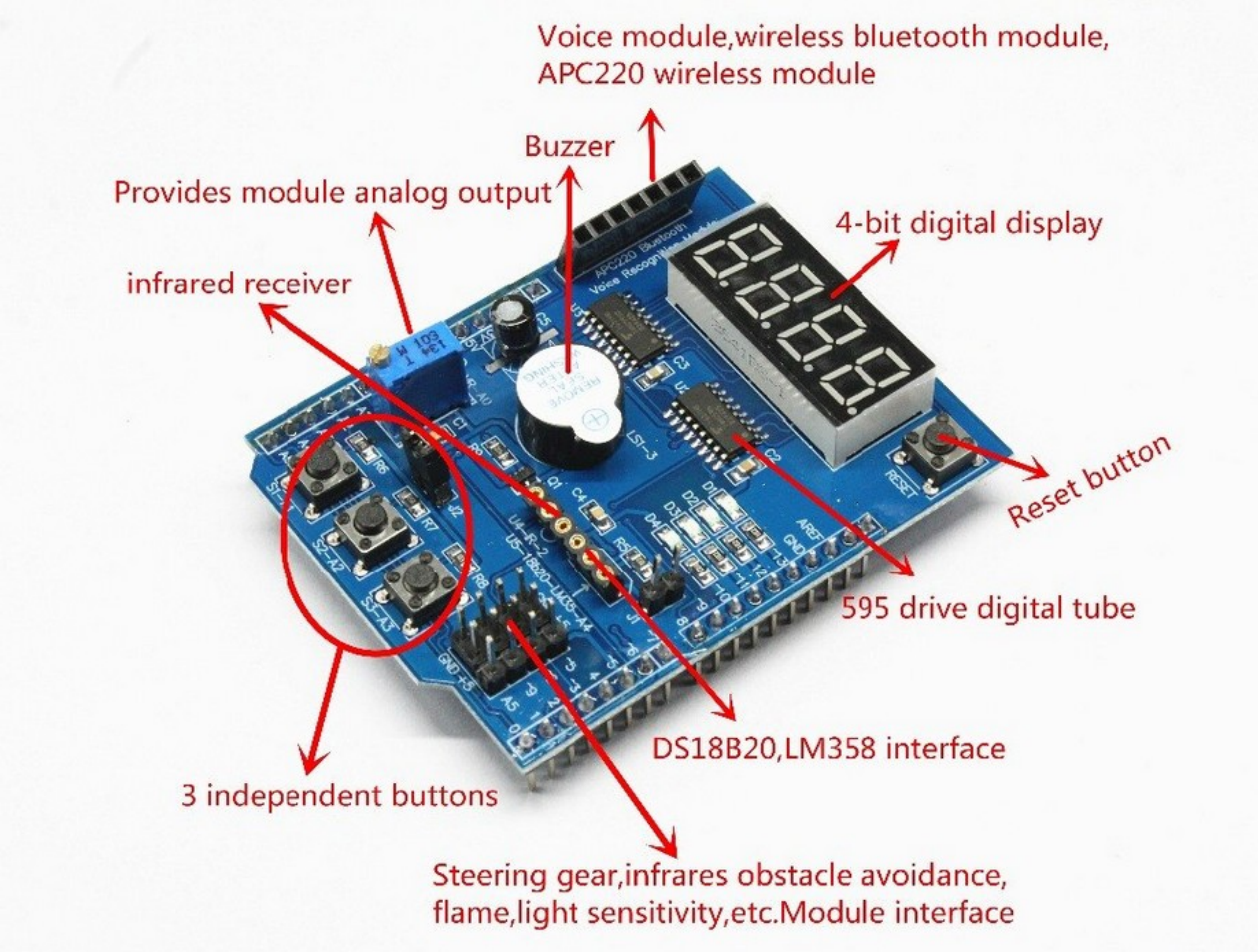
01-SimpleServo | Arduino 1.8.14

File Edit Sketch Tools Help

01-SimpleServo

```
#include <Servo.h>

static const int servoPin = 4;

Servo servo1;

void setup() {
    Serial.begin(115200);
    servo1.attach(servoPin);
}

void loop() {
    for(int posDegrees = 0; posDegrees <= 180; posDegrees++) {
        servo1.write(posDegrees);
        Serial.println(posDegrees);
        delay(20);
    }

    for(int posDegrees = 180; posDegrees >= 0; posDegrees--) {
        servo1.write(posDegrees);
        Serial.println(posDegrees);
        delay(20);
    }
}
```

1 TTGO T-Watch, Enabled, Default (2 x 6.5 MB app, 3.6 MB SPIFFS), 2000000, None on /dev/ttyUSB0


Voice module,wireless bluetooth module,
APC220 wireless module
Buzzer
Provides module analog output
4-bit digital display
infrared receiver
Reset button
595 drive digital tube
DS18B20,LM358 interface
3 independent buttons
Steering gear,infrares obstacle avoidance,
flame,light sensitivity,etc.Module interface

Raspberry Pi

Advantages:

- Very powerful hardware
- Internet access built in (Ethernet and WiFi)
- Runs Linux operating system and can be used as a standalone computer
  - Needs a screen, keyboard and mouse. Interfaces are available
- Programming can be done in any programming language supported by Linux
- Uses micro SD card for OS and storage of local files
- Very popular, with some 10 million units sold
- Good support by community

Raspberry Pi disadvantages

- Not all drivers are available (but most!) for all programming languages
- Higher price (~ 80 US $)

The shopping list

Finally and despite its higher price we went for the Raspberry because of better performance and its use of standard programming languages

We used C for the course.

In addition to the system board we acquired:

- A kit with 37 sensors and actuators
- Stepping motor + controller
- ADC (Analogue to Digital Converter)
- DAC (Digital to Analogue Converter)
- Atmospheric pressure sensor
- 2 line LCD display

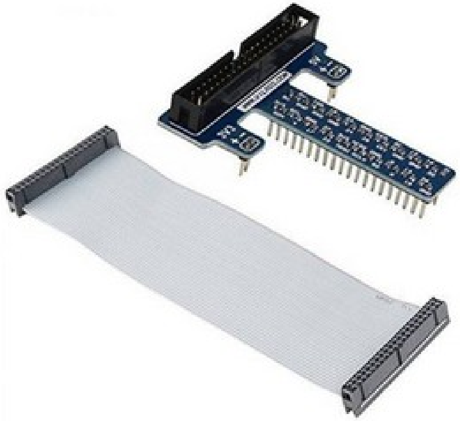
Several Arduinos were bought for student projects.

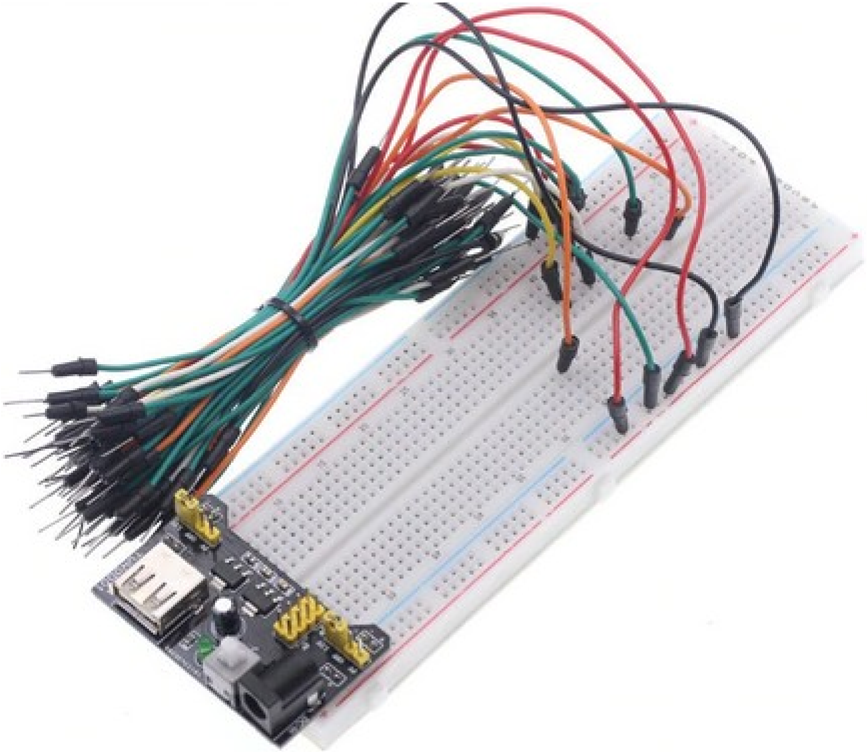
The Raspberry Pi

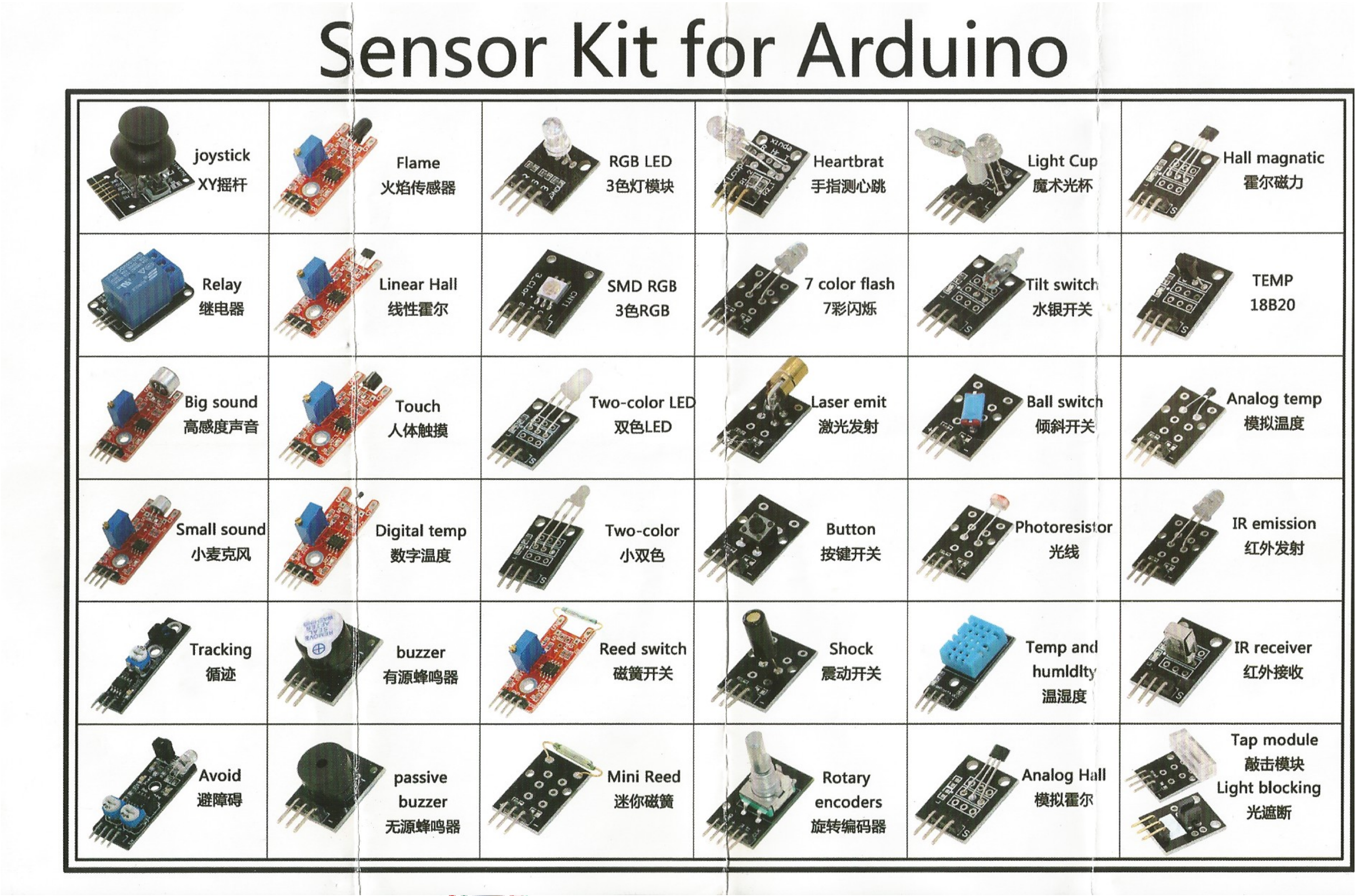
The Raspberry Pi is a small computer powerful enough to run a full blown Linux operating system. The 3rd version of this board features:

- A quad core 1.2 GHz Broadcom BCM2837 64 bit CPU
- I GByte of RAM
- Ethernet and WiFi network interfaces
- 4 USB-2 ports
- Full size HDMI
- a camera port
- Micro SD connector
- A 40 pin extended GPIO connector with GPIO, SPI and I2C bus interfaces

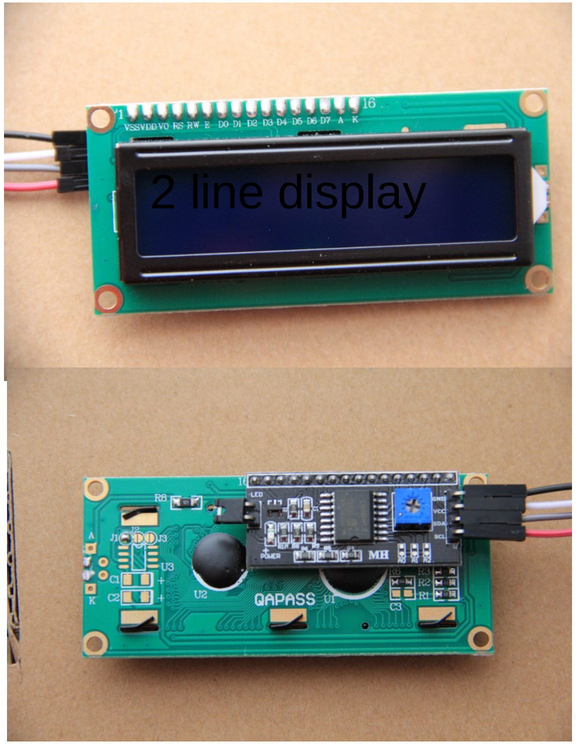
Connecting to the outside world



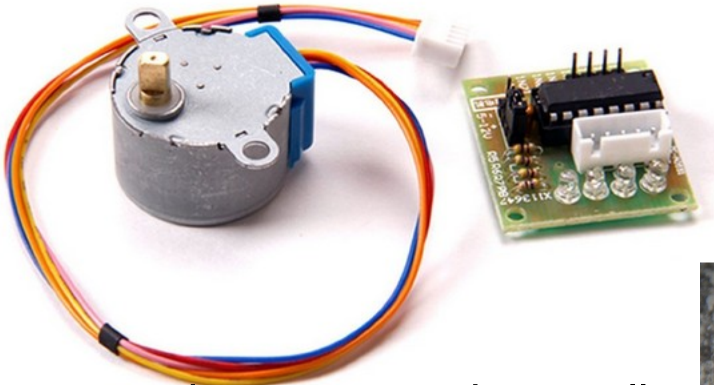



Sensor Kit for Arduino
joystick
XY摇杆
Flame
火焰传感器
RGB LED
3色灯模块
Heartbrat
手指测心跳
Light Cup
魔术光杯
Hall magnatic
霍尔磁力
Relay
继电器
Linear Hall
线性霍尔
SMD RGB
3色RGB
7 color flash
7彩闪烁
Tilt switch
水银开关
TEMP
18B20
Big sound
高感度声音
Touch
人体触摸
Two-color LED
双色LED
Laser emit
激光发射
Ball switch
倾斜开关
Analog temp
模拟温度
Small sound
小麦克风
Digital temp
数字温度
Two-color
小双色
Button
按键开关
Photoresistor
光线
IR emission
红外发射
Tracking
循迹
buzzer
有源蜂鸣器
Reed switch
磁簧开关
Shock
震动开关
Temp and
humldlty
温湿度
IR receiver
红外接收
Avoid
避障碍
passive
buzzer
无源蜂鸣器
Mini Reed
迷你磁簧
Rotary
encoders
旋转编码器
Analog Hall
模拟霍尔
Tap module
敲击模块
Light blocking
光遮断

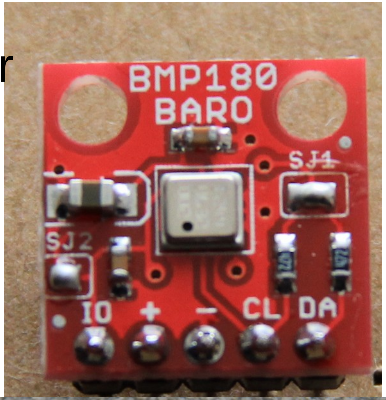
Additional sensors



2 line display

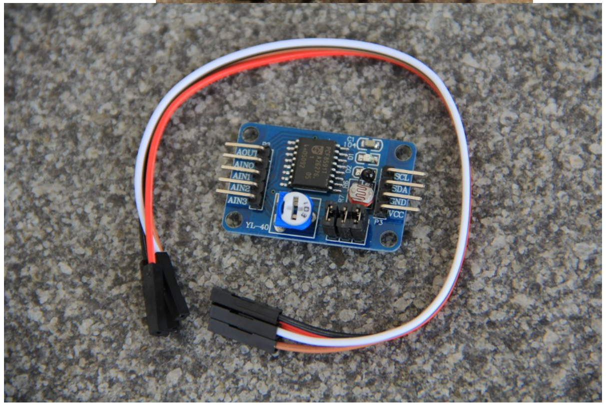


Stepping motor and controller



Barometric pressure sensor

DAC

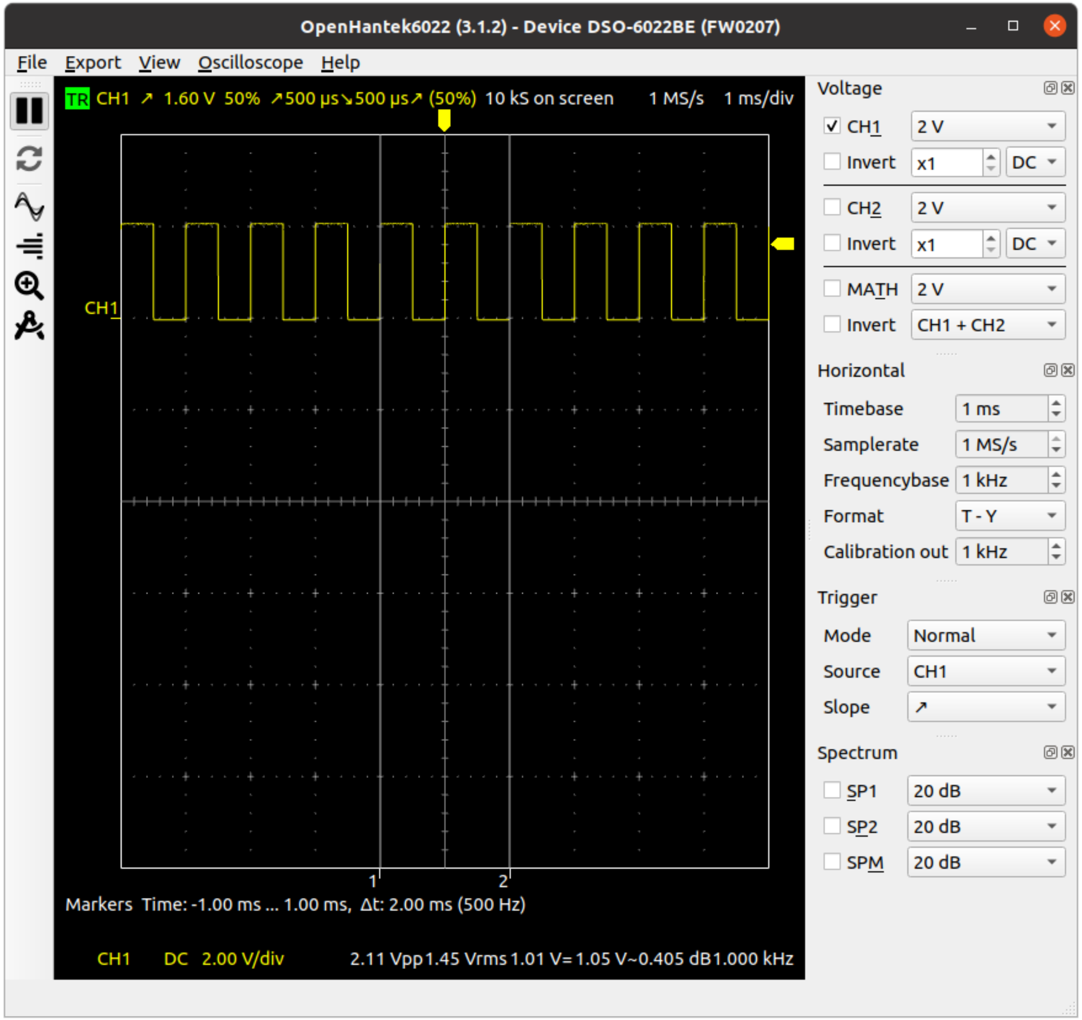


ADC with temperature and light sensor

A cheap oscilloscope

- 2 channels
- 20 MHz bandwidth
- USB interface
- PC used for interaction
- 2 probes
- Only DC coupling
- Cost: 50 Euros

Oscilloscope Software



The course layout

- Introduction
- The Linux operating system and its basic commands
- Introduction to C programming
- An additional lecture on C
- Development tools
- Bringing the Raspberry Pi to life
- Accessing the real world
- DHT11 temperature and humidity measurements
- Stepping motors
- Digital to analogue conversion
- The 2 line display
- Analogue to digital conversion and light intensity measurement

Documentation

- Documentation should be accessible through the WEB
- Should contain
  - Sensor data sheets
  - Lecture slides
  - Exercise sheets and solutions
  - Links to additional documentation
- Should be easy to provide and easy to modify
- Students should be able to provide their own doc

We use TWiki

A difficult start

- The course was foreseen to start in March
- The shopping list was ready in December and I had my kit for preparation in January
- I requested an official written invitation with the conditions
- By mid February no orders were made by the University and I had no invitation. We decided to postpone the course to September
- By June, still no invitation and no order. I set a limit to mid-July because I also had an invitation to lecture at the CERN school of accelerator.
- 2h before the limit I get an email that everything was settled but the department head, responsible to send the invitation would only come the next week.
- It took another 2 weeks and several reminders to get the invitation, but without conditions

A difficult start (2)

- I went to the Ghanaian embassy to get my visa and they asked me if I was sure about the authenticity of the invitation
- 2 weeks before departure, still no hardware order
- I send my flight dates to the university and get an answer that I could only come for 2 months because of budget restrictions. The department head had changed and the new one did not accept the agreements previously made
- I sent an email to Nii and the former department head telling them about the trouble
- The story goes up to the University's Vice chancellor and I finally get the ok for the dates initially foreseen
- Departure, and still no hardware orders

Arrival in Accra

The last two times I was in Africa I was promised to be picked up at the airport and nobody showed up.

This time it **worked!!!**

Ike, the local lecturer (and friend!) and his wife picked me up with a driver from the university and take me to the university guest house in Accra for the night.

Next day we visit Nii at his company premises and continue to Cape Coast

The promised housing is not ready and I am set up in the university guest house at UCC

Meeting again after 20 years

African School of Fundamental Physics and Applications

The department

Preparing the lab

The lab has 20 PCs of which 4 or 5 do not work and the others have totally heterogeneous OS.

We have the permission to install the newest Ubuntu Linux on all machines.

Ike following me and helping me with the preparation

He takes over the lab and the course after my departure

First lecture

- Still no micro-controllers available
- Must start with introduction to Linux and C programming
- Lecture start was scheduled at 9:00 GMT but nobody was there
- I was explained that GMT means Ghana Man Time! So the students would start coming in at ~ 9:30 or 10:00
- Difficult for somebody who worked in Switzerland!
- Students had a 1 semester C++ course but were unable to align 20 lines of C++ code

The micro-controllers and sensors

After the first week we still have no lab! With Ike's and my devices we can equip 2 stations.

The physics department has 5 Raspberry Pis but no power supplies for them! We can borrow them but the Physics department wants another course for its students! This doubles my work load.

A CERN friend sends me this advice:

`No matter how tremendous an obstruction may appear at a distance, you will find that if you go on in a certain way, it will disappear as you approach it, or that a way over, through, or around it will appear`

Oguaa Fetu Afahye

Because of the festival I get kicked out of the guest house and suddenly my room was available

The miracle

After a week and over a beer

Nii promises 1000 US $ to start the lab.

The next day the university releases the budget for the lab.

The supplier in China can deliver immediately and 2 weeks later we have the material.

The lab material

Bring up Raspbian

The operating system distribution is named **Raspbian**, a Debian based Linux distribution for the ARM processor

- The system is distributed in zip format
- Download to the PC and unzip → binary image with 2 partitions
- Copy the image to a micro SD card using Unix *dd*
- Install the micro SD card in the slot on the RPi
- Boot the system
- Upgrade the system to its latest revision with *apt* (Advance Package Tool)
- Run the configurator *raspi-config to* enable drivers for
  - GPIO, I2C, SPI, camera …
- Create new users for the students
- Create an administrator account
- Configure the network
- Do all this with a monitor, USB keyboard and USB mouse connected

Lab preparations

The local area network is available but there are too few network connections to accommodate the Raspberry Pis.

- The switches are available but new network plugs and cables must be installed. Done over the weekend in 2 days!
- The Raspberry Pis need fixed IP addresses to be easily accessible from any PC.
- Raspberry Pis are accessible through ssh or through the remote desktop
- We were working on the configuration literally until seconds before it was needed in the course!

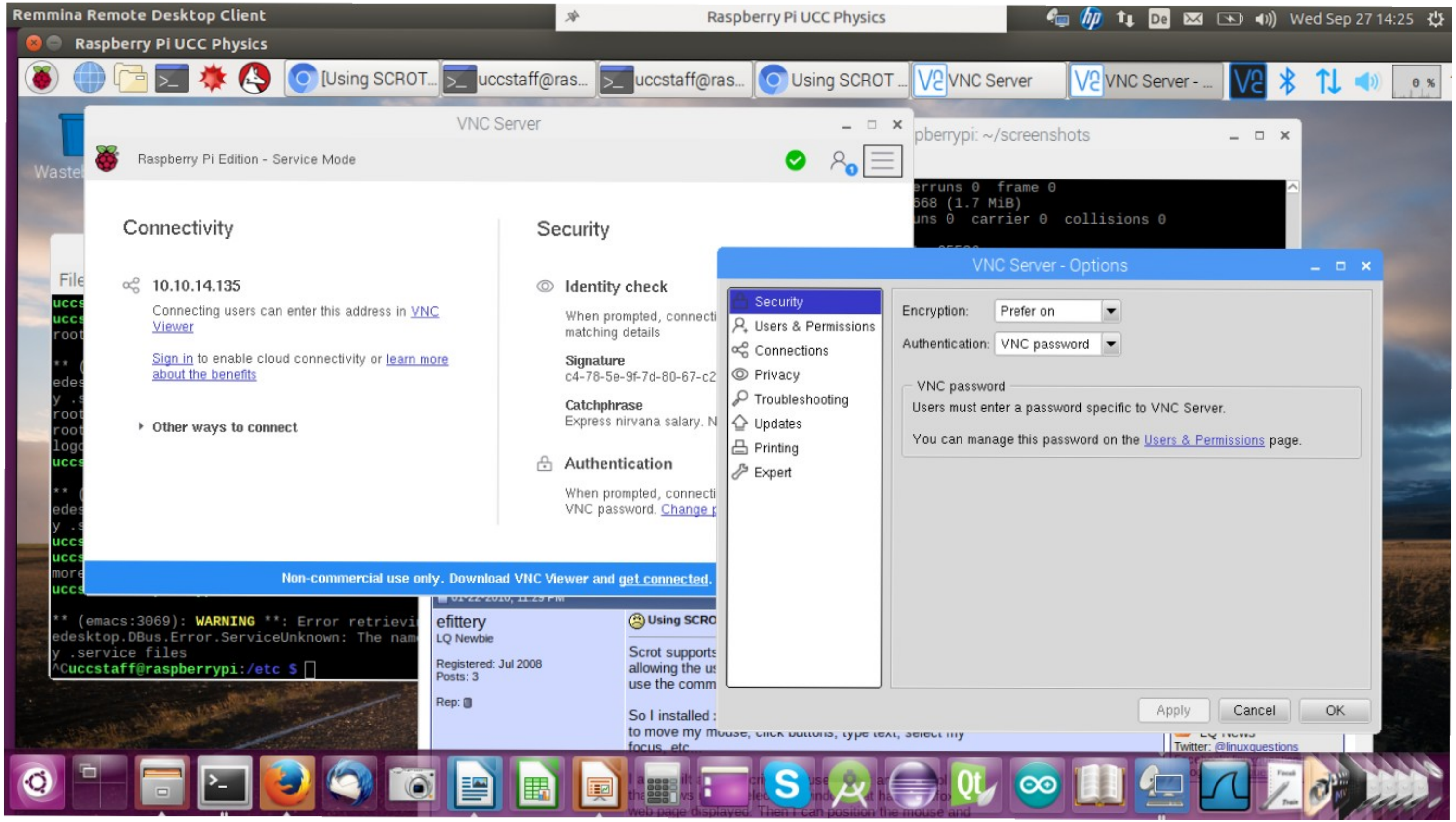
Accessing the Raspberry Pi remotely

Of course we can access the RPI through its screen, keyboard and mouse but this needs a lot of equipment.

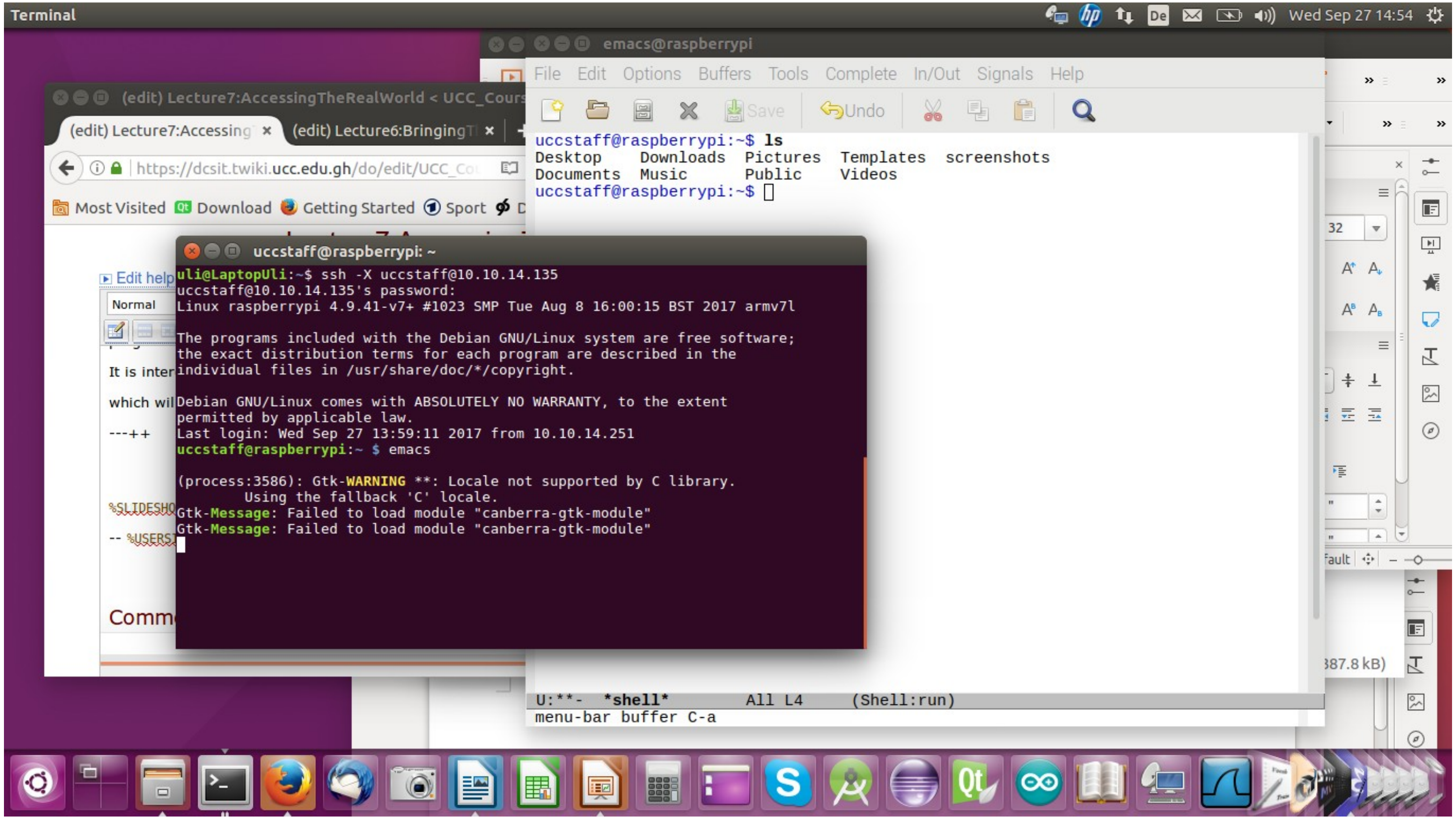
There are several ways to access the RPi remotely:

- Enabling the RPi VNC server and configuring it allows you to access the RPi with a remote desktop client on the PC
- The secure shell ssh allows you to create a remote terminal on the Rpi. With the option ssh -X you enable the X forwarding and you can use the PC as an X-terminal
- With scp you can copy files between PC and RPi
- With NFS you can use part of the PC disk for the RPi.

The remote desktop



An ssh example



First exercise on the RPi

The OS on the RPi looks very similar to Ubuntu on the PC

- The students compile and try their programs on the RPi
- When copying the binary from the PC to the RPi it does not run, why?
- Native vs cross compilations (both are possible on our system)

The “Hello World“ program

Anybody who has learned C will have come across the

“Hello World” program:

```
/*
 * Hello World! the most simple C program you could possibly write
 */
#include <stdio.h>          /* here printf is defined */
int main() {
  printf("Hello World!\n");
}
```

In embedded systems however, this can become quite complex as you may have to configure a UART before being able to send text
and you may gave to set up a serial terminal emulator before being able to receive it.

The “Hello World” equivalent in embedded systems is the blinking LED

Accessing external hardware

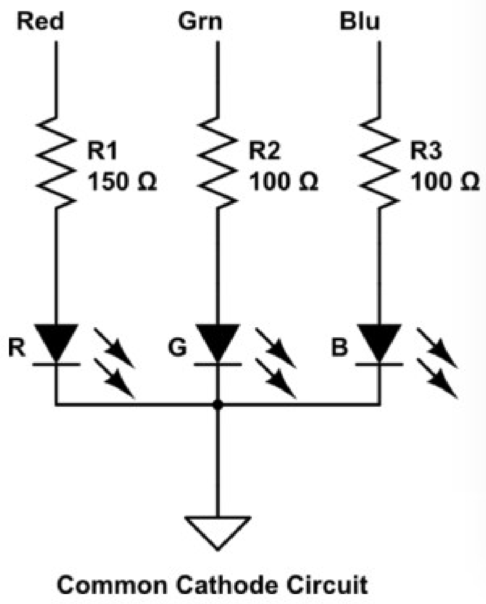
The simplest possible example: blinking a LED

We need to:

- Setup the hardware:
  - Connect the LED cathode to a GPIO pin
  - Connect the anode to ground via a 330 Ω resistor
- Software:
  - Find a hardware access library
  - Program the GPIO pin to be output
  - Write the GPIO pin to switch the LED on
  - Delay the program for 500 ms
  - Write the GPIO pin to switch the LED off

The KY-016 rgb LED

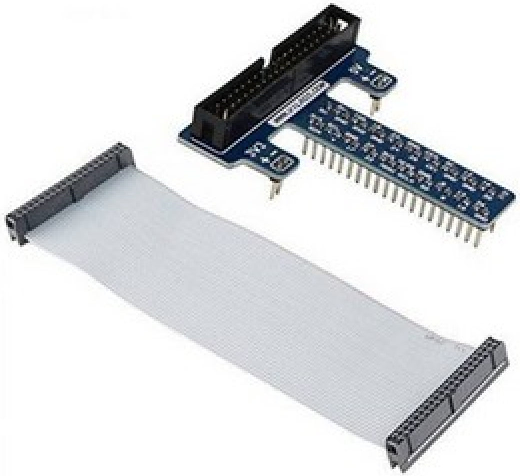
The standard rgb LED on an BY-016 board has the current limiting resistors already on board. The board uses a common cathode circuit, which means that it has a common ground and the rgb signals must be on Vcc to light the LED

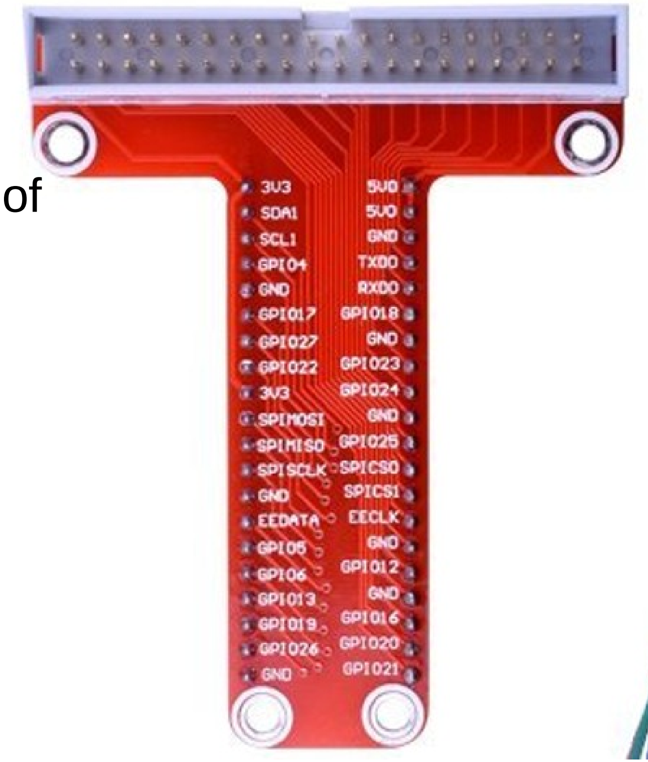


The cobbler

How to make the hardware connections?

The cobbler tells us the signification of the pins





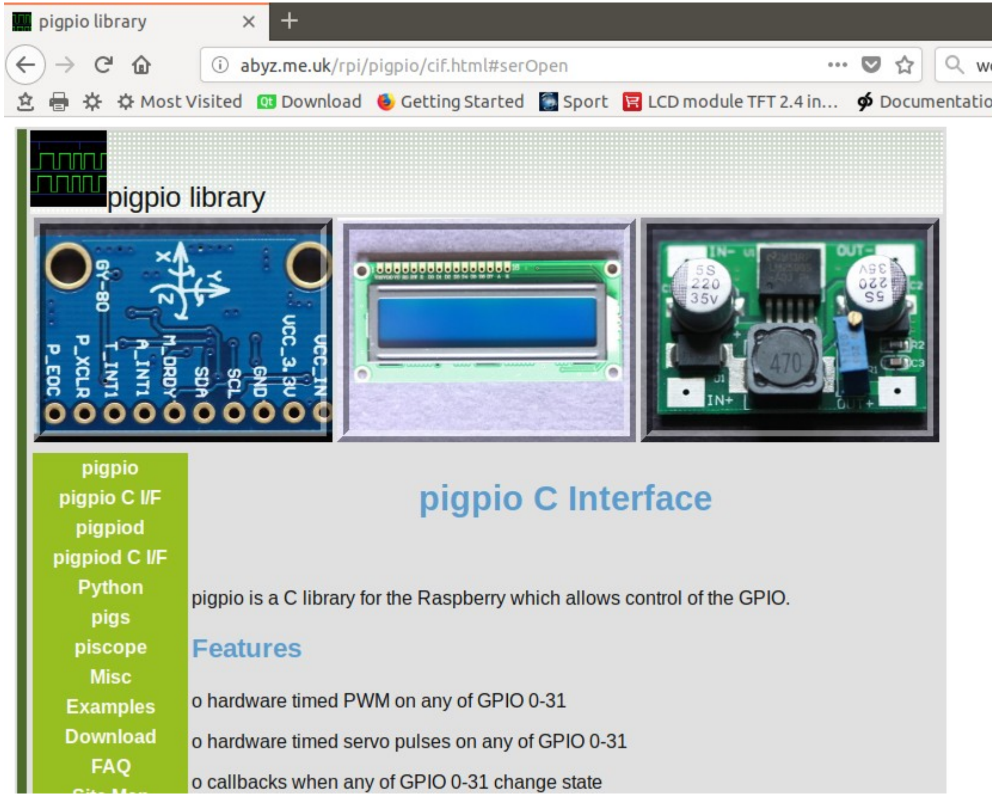
The pigpio library

It is possible to program the GPIO pins accessing the processor registers directly. However this is clumsy and complex.

→ use a library doing the hardware access for us.

pigpio uses a daemon for hardware access. The daemon is started be the super user.

Applications communicate with the daemon through sockets and do not need super user privileges.



Using pigpio

Usage

Include <pigpiod_if2.h> in your source files.

Assuming your source is in prog.c use the following command to build

```
gcc -Wall -pthread -o prog prog.c -lpigpiod_if2 -lrt
```

to run make sure the pigpio daemon is running

```
sudo pigpiod

 ./prog # sudo is not required to run programs linked to pigpiod_if2
```

The library calls for GPIO

All the functions which return an int return < 0 on error

OVERVIEW

ESSENTIAL

pigpio_start	Connects to a pigpio daemon
pigpio_stop	Disconnects from a pigpio daemon

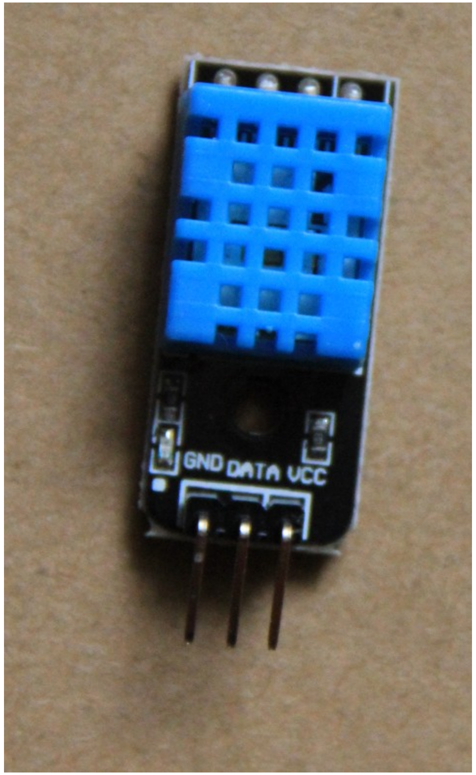
BEGINNER

set_mode	Set a GPIO mode
get_mode	Get a GPIO mode
set_pull_up_down	Set/clear GPIO pull up/down resistor
gpio_read	Read a GPIO
gpio_write	Write a GPIO

blink program

Other GPIO projects

The DHT11 temperature and humidity sensor



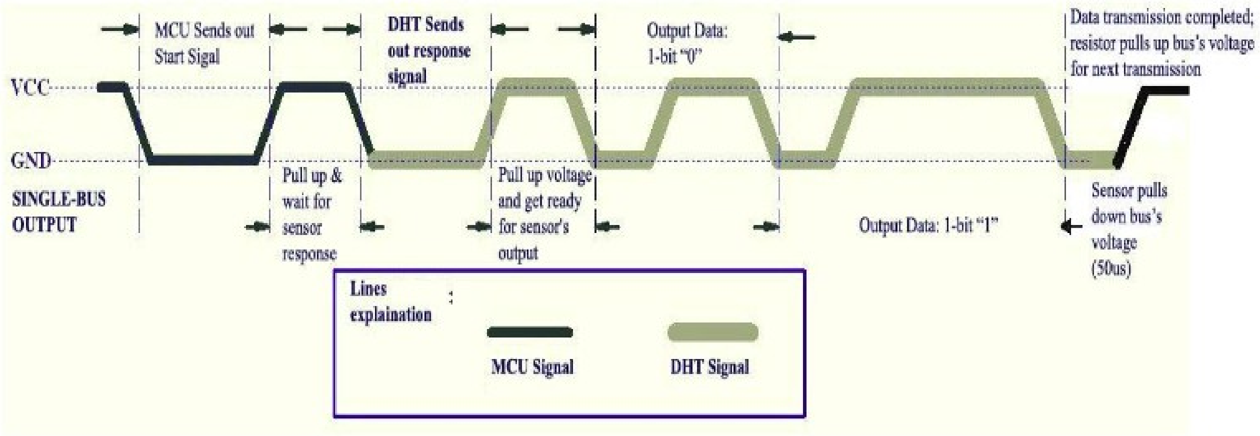
The DHT11 has only
3 pins:

- Vcc
- Ground
- one data pin

Starting a measurement

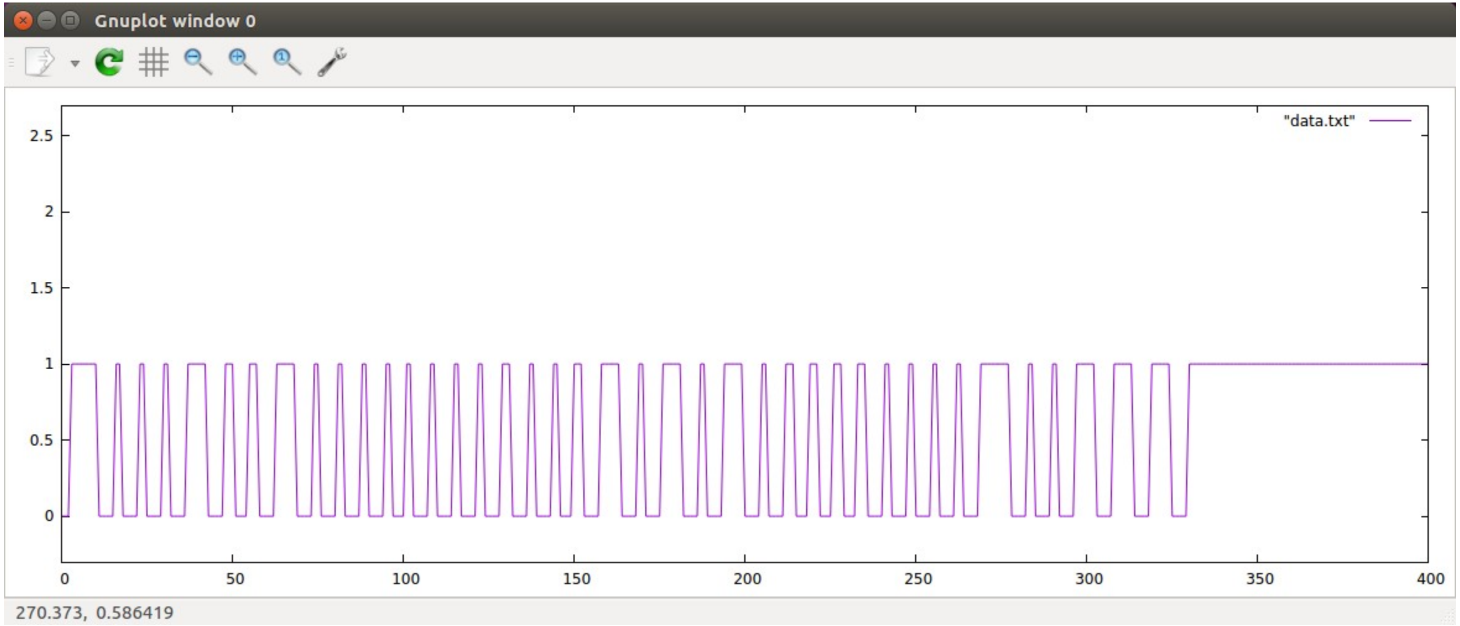
5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

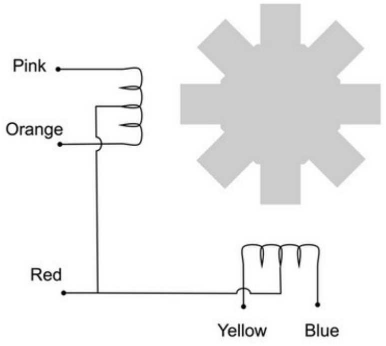


Trigger and Readout

- Program data line to output and send start signal
- Program data line to input and sample data line every 4 µs
- Interpret result

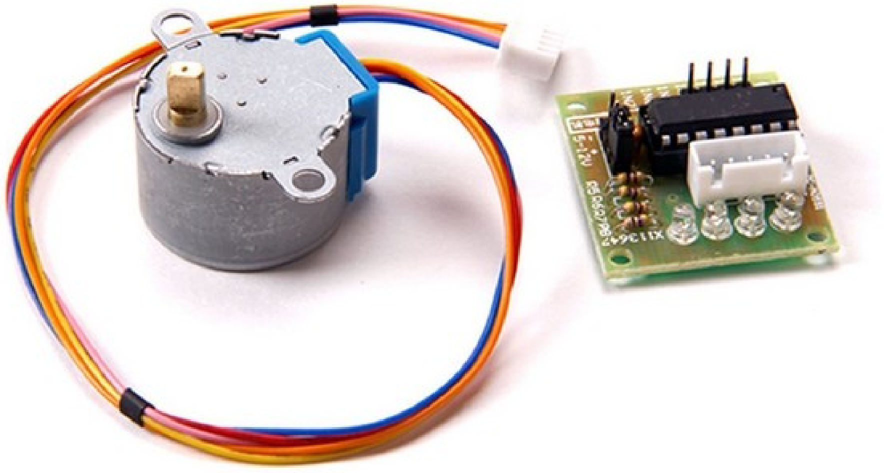


Stepping motor



		Step 1	Step 2	Step 3	Step 4
In 1	coil 1 +	1	0	0	0
In 2	Coil 2 -	0	1	0	0
In 3	Coil 1 -	0	0	1	0
In 4	Coil 2 +	0	0	0	1

Single phase forward stepping



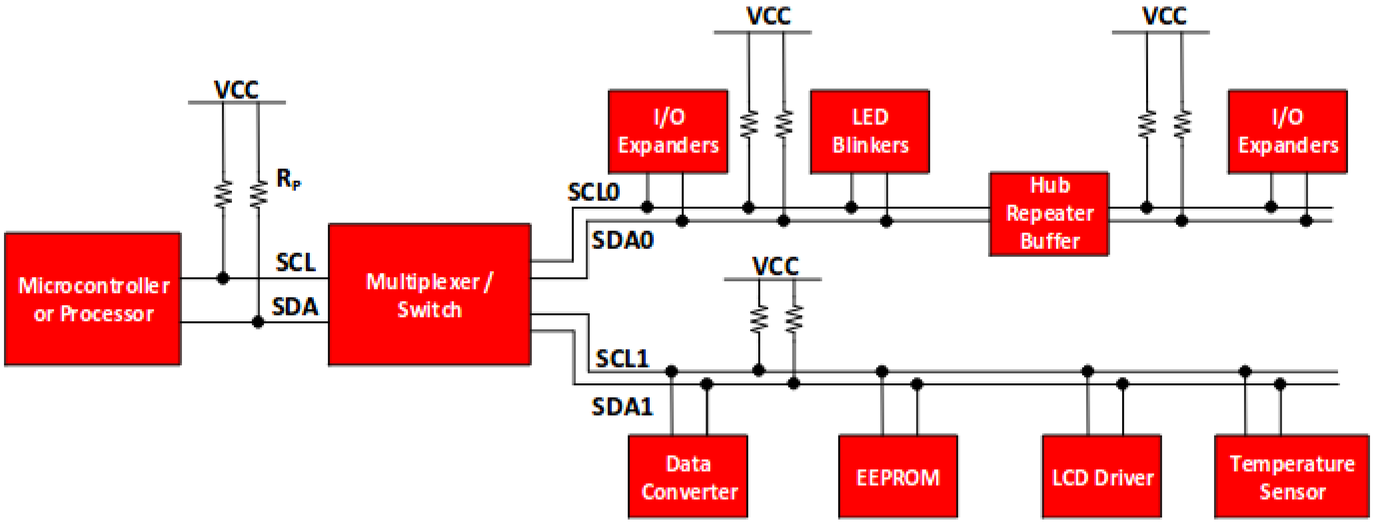
Here we need 4 GPIO signals which must we switched according to he table in order to move the motor by 4 steps

Students at work

The I2C bus

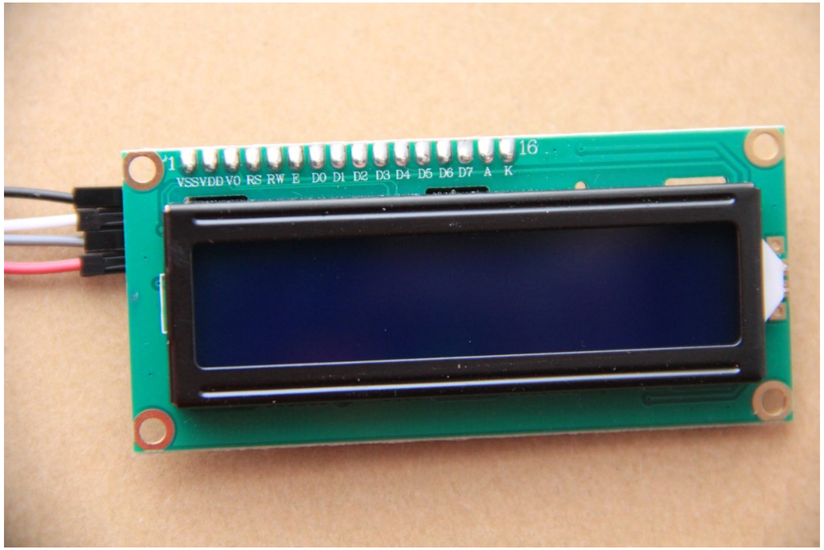
A 2 line serial bus
- SDA: data
- SCL: clock

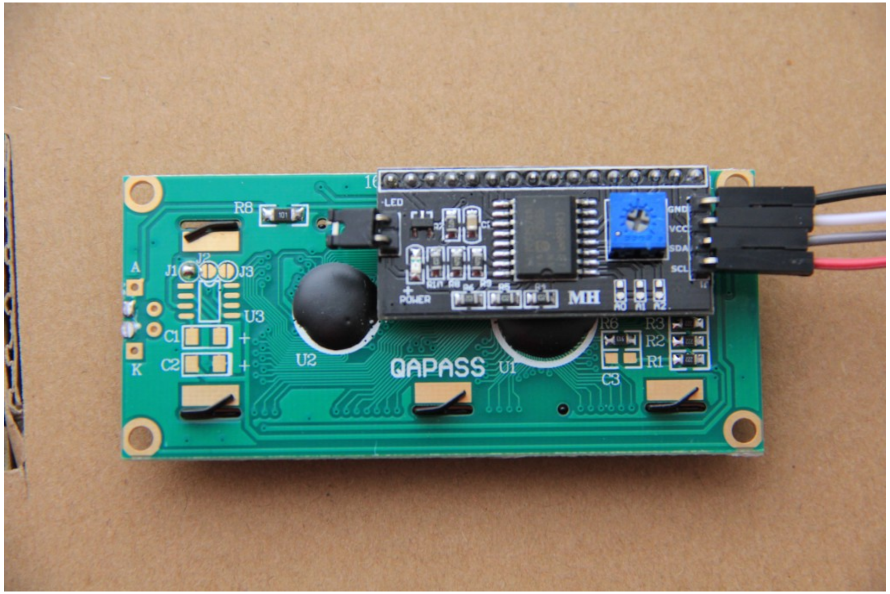
Master sends 7 bit address + R/W followed by data



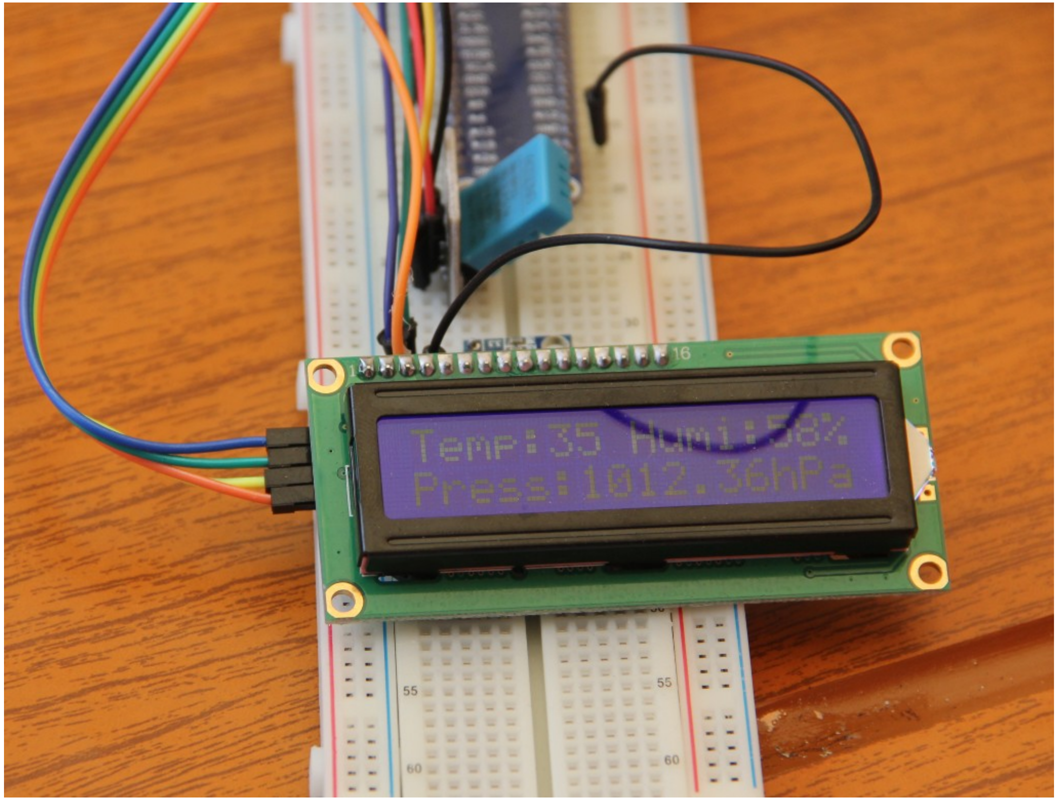
I2C I/O expander

Essentially a shift register allowing to drive a big number of I/O lines with just the 2 I2C signals

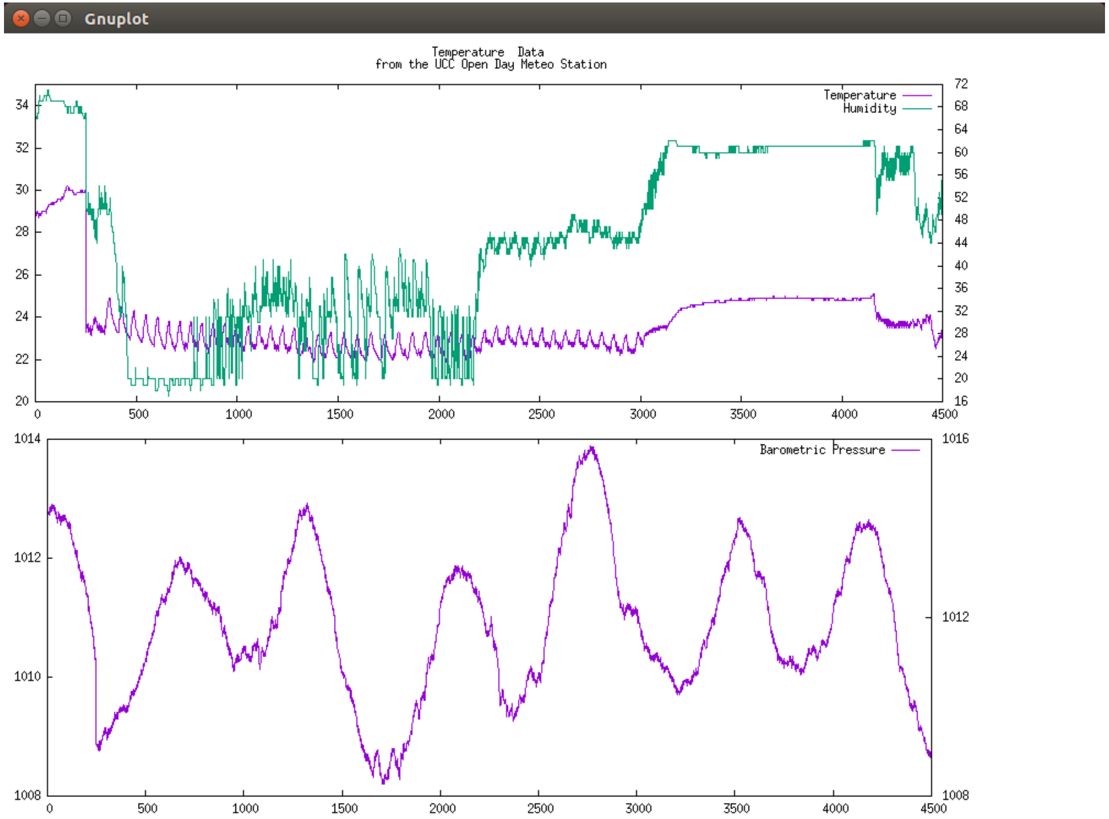




A simple weather station



Results from the weather station



Course exercises on the RPi

- LEDs
  - Blinking, SOS
- Stepping motors (single phase and double phase, half step, forward/backward)
- DHT-11 temperature and humidity measurements
- BMP-180 temperature and barometric pressure measurement
- LCD display driver using an I2C extender
- Digital to analogue converter
  - Create a signal level
  - Create a pulse generator for sine, sawtooth, rectangular and triangular signals
- Analogue to Digital converter: read a potentiometer voltage
- I2C Real time clock
- I2C EEProm

Course exercises on the RPi

- LEDs
  - Blinking, SOS
- Stepping motors (single phase and double phase, half step, forward/backward)
- DHT-11 temperature and humidity measurements
- BMP-180 temperature and barometric pressure measurement
- LCD display driver using an I2C extender
- Digital to analogue converter
  - Create a signal level
  - Create a pulse generator for sine, sawtooth, rectangular and triangular signals
- Analogue to Digital converter: read a potentiometer voltage
- I2C Real time clock
- I2C EEProm

I2C devices in the lab

- 8 bit Analog to digital converter board with
  - Potentiometer
  - Thermistor
  - Photo resistor
  - 1 digital to Analogue channel
- Digital to analogue converter
  - Created a pulse generator
- Real time clock
- EEPROM
  - Programmed some ASCII text and asked the students to find it
- LCD controller interface
- Barometric pressure and temperature sensor

Summary

- After initial difficulties the course went well
- I left a fully functional laboratory for 15 students
- The course was repeated by Ike in 2018 but without any updates
- Passing on experience to the other lecturers did **not** work
  We had lecturer sessions but only a few lecturers came and those came only sporadically
- Ike had the intention to port the whole course to the Python language and during the last 2 weeks I started with that (having no clue about Python programming!) After my own work there was no more progress.
- We wanted to promote the course for other African universities, also without success up to now

The UCC Open Day

The university organizes an Open Day where the students show off projects prepared in the department

The department head asks me to prepare a few presentations

We have:

- A traffic light simulator using LEDs
- Name display: A visitor types his name which is then displayed in a welcome message on an LCD screen attached to the Rpi.
- A stepping motor control program
- A voltmeter
- A small meteorological station measuring
  - Temperature· air pressure, humidity
- A simple obstacle avoiding robot

Photo from the Open Day

The robot

A simple robot:

2 active wheels with DC motors

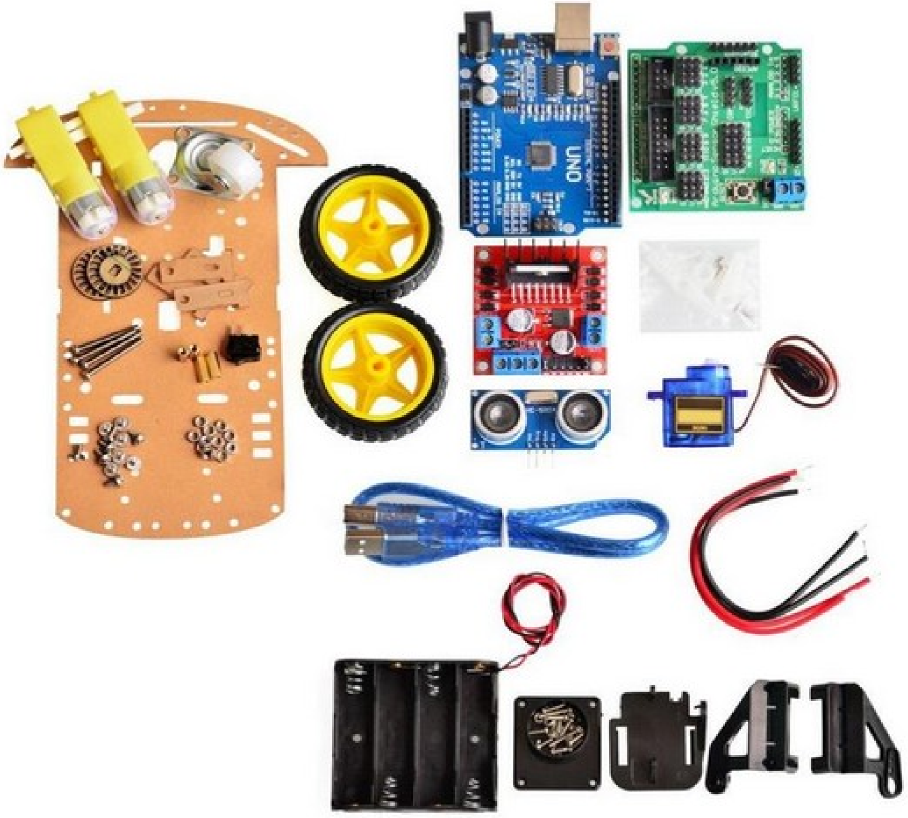
1 passive wheel

Ultra-sonic distance sensor mounted on a

Servo motor

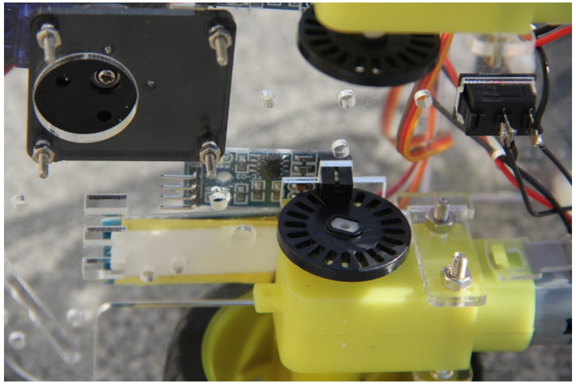
Arduino Uno controller

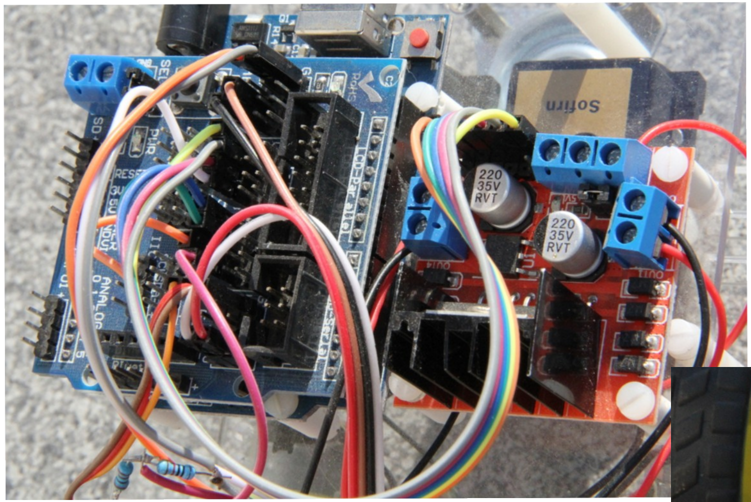
Total cost: < 20 Euros



Robot Details

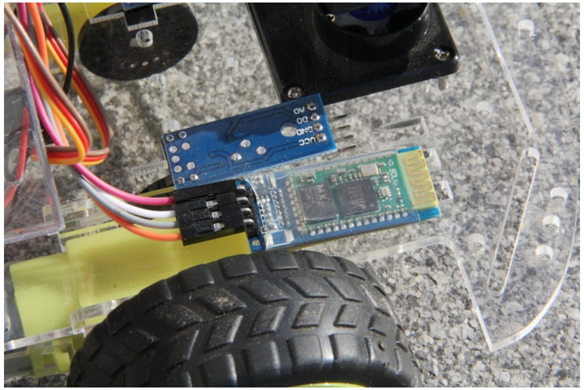
Encoder





Arduino controller and DC motor driver

BlueTooth transceiver



Ultra-sonic distance sensor on servo motor