Embedded Systems

The Arduino and Raspberry Pi HW Platforms
Outline

- Introduction to the Arduino Hardware
- Introduction to single-board Computers
The Arduino Hardware
What is Arduino?

- It’s not a chip, a board or a manufacturer!
- It’s a Open-source movement
  - Open-source hardware platform
  - Open-source development environment
  - Cross-platform (Windows/Mac/Linux)
Why Arduino?

- Easy, fast and inexpensive!
  - Easy programming, uses “Arduino C” which is based on C++
  - Fast prototyping
  - Low-cost HW, or go make yours!
- Perfect for beginners and hobbyists
  - No intensive knowledge in electronics is needed
  - For hobbyists short in time
- Big community
  - Lots of ready-to-use libraries
  - Not only engineers, creativity guaranteed
Why Arduino?

- Many flavors!
  - Uno
  - Mega
  - Leonardo
  - LilyPad
  - Nano
  - Mini
  - BT
What to do with an Arduino?

- Prototyping an idea that compline HW and SW development
  - Network-based coffee machine!
- Interfacing with the physical world
  - Monitor and control
  - Lots of I/O ports
  - A/D and D/A
# The Arduino Uno

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>ATmega328P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limit)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>14</td>
</tr>
<tr>
<td>PWM Digital I/O Pins</td>
<td>6</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>6</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
<td>20 mA</td>
</tr>
<tr>
<td>DC Current for 3.3V Pin</td>
<td>50 mA</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32 KB</td>
</tr>
<tr>
<td>SRAM</td>
<td>2 KB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1 KB</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16 MHz</td>
</tr>
<tr>
<td>Length</td>
<td>68.6 mm</td>
</tr>
<tr>
<td>Width</td>
<td>53.4 mm</td>
</tr>
<tr>
<td>Weight</td>
<td>25 g</td>
</tr>
</tbody>
</table>

![Arduino Uno Diagram]

- **LED**
- **14 digital inputs/outputs (6 PWM outputs)**
- **Power indicator**
- **Reset Button**
- **USB connection**
- **16 MHz clock**
- **Voltage regulator**
- **AC/DC adapter jack**
- **DC voltage supply (IN/OUT)**
- **6 analog inputs**
Arduino programming

- The ‘official’ Arduino programming is done in “Arduino IDE”
  - Programs are called “sketches”
  - Still, you can go with other IDEs, or your own way!
- The IDE:
  - Text editor, with syntax highlighting
  - Compiler
  - HW interface: downloader and serial communication
Arduino sketch

Compile sketch

Upload and run

Serial monitor

Save

Open

New
Arduino sketch

- **Structure**
  - **setup()**
    - Runs once at the start of the program
    - Used for initialization
  - **loop()**
    - Called repeatedly until the board is powered off

- **Example: ‘Blink’**
  - File ➔ Examples ➔ 01.Basics ➔ Blink

```cpp
// the setup function runs once
// when you press reset or power the board
void setup()
{
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over
// and over again forever
void loop()
{
    // turn the LED on (HIGH is the voltage level)
    digitalWrite(13, HIGH);
    // wait for a second
    delay(1000);
    // turn the LED off by making the voltage LOW
    digitalWrite(13, LOW);
    // wait for a second
    delay(1000);
}
```
Where to start?

- Get an Arduino
- Get the IDE
- Design your circuit
- Write your sketch
- Putting all together
- Fix bugs!
‘Design your circuit’ ??

- **Fritzing**
  - Open-source HW prototyping environment
Walkthrough

- Rangefinder
Single-board Computers

Raspberry Pi
Single-board Computers

- A complete computer built on a single board
  - CPU, GPU, RAM, some standard I/O ports
  - Running major OS like Linux and Windows
- Most of them are fanless, reducing power requirements and physical size
- Targets autonomous systems, embedded controllers and low-power applications
- Noticeable examples:
  - Raspberry Pi
Raspberry Pi

- Low-cost, credit-card size single-board computer
- The most famous single-board platform for hobbyists
- Can be used as a standard computer
  - HDMI and USB ports
# Raspberry Pi

## Specs

<table>
<thead>
<tr>
<th>Feature</th>
<th>Model B+</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRCM2835 SoC</td>
<td>Yes</td>
</tr>
<tr>
<td>Standard SoC Speed</td>
<td>700Mhz</td>
</tr>
<tr>
<td>RAM</td>
<td>512MB</td>
</tr>
<tr>
<td>Storage</td>
<td>Micro SD</td>
</tr>
<tr>
<td>Ethernet 10/100</td>
<td>Yes</td>
</tr>
<tr>
<td>HDMI output port</td>
<td>Yes</td>
</tr>
<tr>
<td>Composite video output</td>
<td>On 3.5mm jack</td>
</tr>
<tr>
<td>Number of USB2.0 ports</td>
<td>4</td>
</tr>
<tr>
<td>Expansion header</td>
<td>40</td>
</tr>
<tr>
<td>Number of available GPIO</td>
<td>26</td>
</tr>
<tr>
<td>3.5mm audio jack</td>
<td>Audio/Video</td>
</tr>
<tr>
<td>Number of camera interface ports (CSI-2)</td>
<td>1</td>
</tr>
<tr>
<td>Number of LCD display interface ports (DSI)</td>
<td>1</td>
</tr>
<tr>
<td>Size</td>
<td>85 x 56 x 17mm</td>
</tr>
</tbody>
</table>
OS support

- **Primary:**
  - Raspbian
  - Debian Wheezy Linux
  - New Out Of the Box Software (NOOBS)

- **Others:**
  - RISC OS
  - Arch Linux
  - Pidora (Fedora Remix Linux)
  - Occidentallis (Raspbian + preinstalled GPIO etc)
  - Media Centre
  - Windows IoT
Getting started

You’ll need:
- Raspberry Pi!
- SD card (>= 16 GB, class 10 preferred)
- Power supply (5 volt, 1200 amp)
- HDMI-capable screen, or LCD with touch
- USB mouse and keyboard (wireless is better)
- Wifi nano adapter (or Ethernet cable)
RPI Programming

- Most of the RPI developers rely on Python
  - Using C programs is used too
  - Native and cross-compiled

- Other important packages
  - HTTP web server → apache
  - Database → MySQL
  - Message-queue handlers → MQTT

- Interfacing
  - GPIO
  - SPI
  - I²C
  - UART
  - Network sockets

- Fritzing!
Walkthrough

- RPI SPI communication with STM32F