MICROCONTROLLERS

Maurício Féo
m.feo@cern.ch
CERN/Nikhef
OBJECTIVES

▶ Understand what are microcontrollers.
  ▶ What are they?
  ▶ What are they used for?
  ▶ How do they work?
  ▶ How can I use them?
  ▶ Are they suitable for my project?

▶ Example of applications.
▶ Have an overview of the lab 10.
WHAT IS A MICROCONTROLLER?

Tiny computer integrated in the same chip.

- CPU
- Memories
- I/O Interfaces
- Etc.

Maurício Féo – ISOTDAQ 2019
WHAT IS A MICROCONTROLLER?

- Tiny computers integrated in a single chip
  - CPU, Memories and Peripherals in the same chip.

Main differences w.r.t. a computer:
- Suitable for embedded applications.
- Low cost (ATtiny: ~$0.27)
- Low power consumption (ATtiny43U: 0.15uA in Sleep Mode)
- Reduced clock frequency (~ dozens of MHz)
- Stand-alone devices (Some require only power to work)
- Low level control of your application.

Maurício Féo – ISOTDAQ 2019
WHAT ARE THEY USED FOR?

- Monitoring
- Data Acquisition
- Control

Applications where:

- High performance is not required.
- Other devices are inadequate (overkill) due to:
  - High power consumption
  - need of external memories and peripherals
  - cost
  - etc.
WHERE ARE THEY USED?

- Everywhere!
- Consumer electronics, home appliances, toys, vehicles, computers, hobbyist projects, etc.

- According to Wikipedia trusted sources, a typical mid-range automobile has as many as 30 or more microcontrollers.
- According to me an even more trusted source, you have at least one in your pocket right now.
An error has occurred. To continue:

Press Enter to return to Windows, or

Press CTRL+ALT+DEL to restart your computer. If you do this, you will lose any unsaved information in all open applications.

Error: 0E : 016F : BFF9B3D4

Press any key to continue
An error has occurred. To continue:

Press Enter to return to Windows, or

Press CTRL+ALT+DEL to restart your computer. If you do this, you will lose any unsaved information in all open applications.

Error: 0E : 016F : BFF9B3D4

Press yes to continue.
AVR ARCHITECTURE (ATMEGA328P)

The one used on the lab.
AVR ARCHITECTURE (ATMEGA328P)

The one used on the lab.
AVR ARCHITECTURE (ATMEGA328P)

- 8 bits architecture
- 32kB Flash program memory
- 2kB RAM
- 2kB EEPROM
- Max 20MHz
- 6 x PWM
- 6 x ADC channels (10bits)
- 23 I/O pins
- 3 timers (2x8 bits 1x16 bits)
- 1x USART
- 1x SPI
- 1x TWI (I²C)
- 0.6mA/MHz

The one used on the lab.
AVR ARCHITECTURE (ATMEGA328P)

Maurício Féo – ISOTDAQ 2019
AVR CPU

- Harvard Architecture

- 8 bits architecture (with 16 bits for instructions)
  - Instructions executed 8 bits by 8 bits

- Reduced Instruction Set Computing (RISC) (~130 instructions)

- Up to 20 MIPS at 20 MHz (1 instruction / clock cycle)
AVR ARCHITECTURE (ATMEGA328P)
GENERAL PURPOSE INPUT/OUTPUT (GPIO)

- Pins programmable as Input and Output
- Read / Write digital signals
- ‘0’ = 0V (Gnd), ‘1’ = 5V (Vcc)

- Controlled by 3 registers:
  - DDR (Data Direction Register)
  - PORT (Where you write when it’s output)
  - PIN (Where you read when it’s input)

Maurício Féo – ISOTDAQ 2019
INTERRUPT

- Interrupts break the program flow to handle some event.

- It may be triggered by:
  - Pin change (rise/fall/toggle)
  - Timers / Counters
  - Analog Comparator
  - ADC reading done
  - Serial interfaces (Rx/Tx done)

- It allows the program to handle an event "right after" its occurrence, regardless of where the program is and without the need of polling constantly.

<table>
<thead>
<tr>
<th>Main program</th>
</tr>
</thead>
<tbody>
<tr>
<td>instruction 1</td>
</tr>
<tr>
<td>instruction 2</td>
</tr>
<tr>
<td>instruction 3</td>
</tr>
<tr>
<td>instruction 4</td>
</tr>
<tr>
<td>instruction 5</td>
</tr>
<tr>
<td>instruction 6</td>
</tr>
<tr>
<td>instruction 7</td>
</tr>
<tr>
<td>instruction 8</td>
</tr>
<tr>
<td>instruction 9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISR()</th>
</tr>
</thead>
<tbody>
<tr>
<td>inst 1</td>
</tr>
<tr>
<td>inst 2</td>
</tr>
<tr>
<td>inst 3</td>
</tr>
<tr>
<td>inst 4</td>
</tr>
</tbody>
</table>
TIMERS / COUNTERS

- Internal registers that increment triggered by:
  - A clock source: **Timer**
  - An external event: **Counter**
- May be used to:
  - Measure time
  - Raise interruption on:
    - Overflow
    - Reach a certain value (**OCR**)
  - Create waveform
    - PWM

Ultrasonic distance sensor
Measures distance based on the time to echo of an ultrasonic pulse.

Maurício Féo – ISOTDAQ 2019
PULSE WIDTH MODULATION (PWM)

- You can create an output signal which value depends on the status of the timer.
- Outputs a train of periodic digital pulses with controlled width.
  - (Can be used to "mimic" an analog signal)
PULSE WIDTH MODULATION (PWM)

- You can create an output signal which value depends on the status of the timer.
- Outputs a train of periodic digital pulses with controlled width.
  - (Can be used to "mimic" an analog signal)
AVR ARCHITECTURE (ATMEGA328P)
ANALOG COMPARATOR

- Tells whether positive pin AIN0 voltage is higher than negative pin AIN1.
- Output is the internal bit ACO* of reg ACSR*.
- Can be used to:
  - Compare two analog signals
  - Trigger a Timer/Counter
  - Trigger an interrupt (rise, fall or toggle)

\[
V_1 > V_2 \quad \Rightarrow \quad V_{out} = 1 \\
V_1 < V_2 \quad \Rightarrow \quad V_{out} = 0
\]

*ACO = Analog Comparator Output
*ACSR = Analog Comp. Control and Status Register
ANALOG TO DIGITAL CONVERTER (ADC)

- 10-bit resolution
  - 0V-Vref → 0-1023
- Vref can be:
  - Vcc (Power source)
  - 1.1V internal ref. (from bandgap)
  - External ref. on pin ‘AREF’
- Successive approximation
  - 13-260 us Conversion time
- Interrupt on complete
- 6 multiplexed channels on DIP package
  - (internal Temp sensor on ch8)
AVR ARCHITECTURE (ATMEGA328P)

Maurício Féo – ISOTDAQ 2019
SERIAL INTERFACES: USART
UNIVERSAL SYNCHRONOUS-ASYNCHRONOUS RECEIVER TRANSMITTER

- A simple protocol
- Widely used to communicate with PCs due to compatibility with RS232 protocol. (RS232 is not used anymore in most PCs but it’s still very easy to find USB-Serial converters)
- Up to 250kbps
- May trigger interrupts:
  - Tx complete
  - Rx complete
  - Data reg empty

Maurício Féo – ISOTDAQ 2019
Differently from the USART, SPI can talk to multiple devices on the same bus, but needs a Slave Select signal per Slave Device.

- Up to 10Mbps! (clk/2)

- Slaves do not "talk" autonomously.
  - Must be queried (and clocked) by master.

All SD cards use the SPI serial interface and can be easily accessed from a uC.
SERIAL INTERFACES: TWI (I²C)
TWO WIRE INTERFACE (INTER-INTEGRATED CIRCUIT)

- I²C allows multiple Masters and Slaves on the same bus. (up to 128)
- Up to 400kbps (on the Atmega328)
- Used in a variety of digital sensors.
AVR ARCHITECTURE (ATMEGA328P)

Maurício Féo – ISOTDAQ 2019
WATCHDOG TIMER (WDT)

- A Watchdog Timer is a timer clocked by an on-chip oscillator.
- Once the counter reaches a certain value, the microcontroller may:
  - Trigger an interrupt
  - Reset the microcontroller
- Used to prevent your program from getting stuck in any part of the code.
- You use it by enabling the WDT and spreading WDT reset instructions on particular places of your code.
  - If it gets stuck in an infinite loop, for ex., the counter won’t be reset and the microcontroller will be reset.
CLOCK CIRCUIT

- Up to 20MHz from:
  - External clock from a pin
  - External crystal oscillator
  - Internal RC oscillator
  - 7.3-8.1 MHz
  - 128kHz Internal oscillator
  - 128 kHz

- System Clock Prescaler
  - Divides the clock if needed

- Keep in mind:
  - Power consumption is proportional to clock frequency.
AVR ARCHITECTURE (ATMEGA328P)
There are multiple Sleep Modes available. Each turns off certain parts of the microcontroller to save power and can only be woken up by certain sources.

### Table: Sleep Modes

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Condition</th>
<th>Min.</th>
<th>Typ. (2)</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power-down mode (3)</td>
<td>WDT enabled, $V_{CC} = 3V$</td>
<td>4.2</td>
<td>8</td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td>WDT disabled, $V_{CC} = 3V$</td>
<td>0.1</td>
<td>2</td>
<td></td>
<td>μA</td>
</tr>
</tbody>
</table>

### Table: Active Clock Domains

<table>
<thead>
<tr>
<th>Sleep Mode</th>
<th>clk_cpu</th>
<th>clk_Lash</th>
<th>clk_io</th>
<th>clk_adc</th>
<th>clk_asy</th>
<th>Main Clock Source Enabled</th>
<th>Timer Oscillator Enabled</th>
<th>INT1, INT0 and Pin Change</th>
<th>TWI Address Match</th>
<th>Timer2</th>
<th>SPI/EEPROM Ready</th>
<th>ADC</th>
<th>WDT</th>
<th>Other I/O</th>
<th>Software BOD Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X(2)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ADC Noise Reduction</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X(2)</td>
<td>X(3)</td>
<td>X</td>
<td>X(2)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Power-down</td>
<td>X(3)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Power-save</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Standby</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Extended Standby</td>
<td>X(2)</td>
<td>X</td>
<td>X(2)</td>
<td></td>
<td></td>
<td>X(3)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
POWER AND DEBUG

- Brown-Out Detector (BOD)
  - Resets the device whenever Vcc is below a certain threshold.
- Power-on Reset (POR)
  - Ensures the device is reset from Power On.
- DebugWIRE
  - On-chip debug tool from AVR.
REVIEW

Maurício Féo – ISOTDAQ 2019
ATMEGA328 MINIMUM REQUIRED CIRCUIT
NEEDED CIRCUITRY: USING AN EXTERNAL CRYSTAL

1: 100 nF
2: 18~22 pF
R: 10k
Cristal: 16MHz

Máx: 5.5V
Mín: 3.8V

Maurício Féo – ISOTDAQ 2019
ATMEGA328 MINIMUM REQUIRED CIRCUIT
NEEDED CIRCUITRY: USING THE INTERNAL OSCILLATOR

- Atmega328 comes with an internal 8MHz oscillator that can minimize the required circuit to a single battery.
USAGE OF MICROCONTROLLERS
DEVELOPMENT CYCLE

► Write your code. From Assembly to C. Or even higher level:

► Compile it. (debug)

► Upload to the uC memory (On Chip debug)
  ► Parallel programming
  ► Serial downloading (SPI)
  ► Self-programming (With bootloader)

► (Burn the fuses)
The AVR core can write to its own program memory.

The Bootloader Section can be locked from rewriting.

This way developers can allow users to write their own programs without compromising the bootloader section.

This can also be used to ease the programming of the memory, eliminating the need for an external programmer.
ARDUINO

Maurício Féo – ISOTDAQ 2019
ARDUINO

- Open-source platform created for "makers" who have no knowledge in electronics but still want their creations to interact with the environment.
- Custom IDE + libraries
- Inexpensive and really easy to use
  - (Almost plug and play)
- Huge community all over the world creating libraries, compatible hardware and sharing projects
- Stackable addons called shields
SETTING PWM TO 25% DUTY CYCLE
THE MEDIEVAL ORIGINAL WAY

Registers:

- **DDR** = Data Direction Register
- **TTCR0A** = Timer/Counter Control Register A
- **TTCR0B** = Timer/Counter Control Register B
- **OCR0A** = Output Compare Register 0 A

```
GPIO D6 PORTD = 0x40;
TCCR0A |= (1<<WGM00) |(1<<WGM01);
TCCR0A |= (1<<COM0A1)|(1<<COM0A0);
OCR0A = 0x3F;
TCCR0B |= (1<<CS00);
```

Bits:

- **WGM0[1..0]** = Waveform Generator Mode 0
- **COM0A[1..0]** = Compare Output Mode 0 A
- **CS0[2..0]** = Clock Select 0

Set direction of pin 6 from PORTD to output.
Set WGM to Fast PWM.
Set COM to «Clear On Compare».
Set OCR to 25% (0x3F of 0xFF)
Set the clock source to timer.
SETTING PWM TO 25% DUTY CYCLE
THE CHEATING ARDUINO WAY

Look at the board which pin you want to use.

```
analogWrite(6, 63);
```

\[ \frac{25}{100} = \frac{x}{255} \]

Find x.
ADVANTAGES OF PROTOTYPING PLATFORMS

NEVER HAVE I FELT SO CLOSE TO ANOTHER SOUL
AND YET SO HELPLESSLY ALONE
AS WHEN I GOOGLE AN ERROR
AND THERE’S ONE RESULT
A THREAD BY SOMEONE
WITH THE SAME PROBLEM
AND NO ANSWER
LAST POSTED TO IN 2003

WHO WERE YOU, DENVERCODER?!
WHAT DID YOU SEE?!

http://xkcd.com/979/
### ADVANTAGES OF PROTOTYPING PLATFORMS

<table>
<thead>
<tr>
<th>PRO</th>
<th>CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community / Support / StackOverflow</td>
<td>Performance is generally not good</td>
</tr>
<tr>
<td>Much easier to learn</td>
<td>No full control over the code</td>
</tr>
<tr>
<td>Fast development and Prototyping</td>
<td>Cost is higher</td>
</tr>
<tr>
<td>Portable code between supported devices</td>
<td>TRU ENGINEERS gon’ make fun of u cos ur not BRAVE ENOUGH to handle raw bits.</td>
</tr>
</tbody>
</table>
Is a microcontroller suitable for my application?

- Cost
- Development time
- Power consumption
- Processing power
- Timing requirements
- Etc.
USAGE OF MICROCONTROLLERS
CHOOSING A MICROCONTROLLER

- What kind of problem do I have?
  - Processing intensive? Power limitation? Embedded?
- Which kind of sensors/actuators will be used?
  - Digital/Analog? Voltage levels?
- What are the required peripherals?
  - USB? I2C? ADCs? Timers?
- What is the environment?
  - Space? Right next to the LHC beam?

Alternatives?
EMBEDDED SYSTEMS

- Microcontroller
- FPGA – Field Programmable Gate Array
- DSP – Digital Signal Processor
- SoC – System On a Chip
- Single Board Computer
- ASIC – Application Specific Integrated Circuit
EMBEDDED SYSTEMS

FPGA Xilinx Spartan 6
Ethernet Controller + USB

Raspberry Pi Zero
SoC (no FPGA)
$5 USD
1GHz
512MB RAM

Raspberry Pi 3
1.2GHz 64-bit quad-core
1GB RAM
Wifi + Bluetooth

NodeMCU
~$10 USD
Wifi Microcontroller
APPLICATIONS IN HEP ENVIRONMENTS

Arduino as a delay control unit, as a temperature monitor,
APPLICATIONS IN HEP ENVIRONMENTS

Arduino as a delay control unit,
as a temperature monitor,
as an USB TTL/NIM generator.
APPLICATIONS IN HEP ENVIRONMENTS
APPLICATIONS IN HEP ENVIRONMENTS

Arduino as a remote GPIB controller.
APPLICATIONS IN HEP ENVIRONMENTS

Telegram BOT for Testbeam monitoring.

What's your purpose?
-To report status and wake up Mauricio at 3am.

November 03, 2017
M. Féo
APPLICATIONS IN HEP ENVIRONMENTS

Is it possible to build a complete particle detector and data acquisition system using Arduino microcontroller and Arduino Language?
Applications in HEP Environments

ArduSiPM a low cost particle detector

http://www.arduino.org/blog/ardusipm-solution

“The ambit of data acquisition for particle detection is a field apparently limited to top scientists from CERN in Geneva and Fermilab in Chicago. Cosmic ray and radiation detection can be a great exploration for teachers, students and science enthusiasts, and ArduSiPM was created to make it accessible.”

Cosmic Ray detector

[Diagram showing a Cosmic Ray detector with a Particle Detector, Arduino DUE, ArduSiPM Shield, and a WIFI module.]
APPLICATIONS IN HEP ENVIRONMENTS

Application Example 2:
Use of ArduSiPM in the CERN UA9 and CRYSBEAM activity
(substitute old Scintillator and electronics for PM)

- As beam trigger @ extracted beam line H8 (CERN)
- As beam losses counter @ SPS

This work has been supported by the ERC Ideas Consolidator Grant
No.615089 “CRYSBEAM”.
LAB 10 - MICROCONTROLLERS
THAT’S IT.

OBRIGADO!

Maurício Féo
m.feo@cern.ch
Figure 13-2. General Digital I/O(1)

GPIO DIAGRAM
Figure 23-1. Analog to Digital Converter Block Schematic Operation,

Maurício Féo – ISOTDAQ 2019
ANALOG COMPARATOR
Figure 19-1. USART Block Diagram\(^{(1)}\)