Republic of Tunisia
Ministry of Higher Education and Scientific Research
University of Tunis El Manar
National Engineering School of Tunis

Presentation

ATmega328P Register Level Programming with Embedded Rust

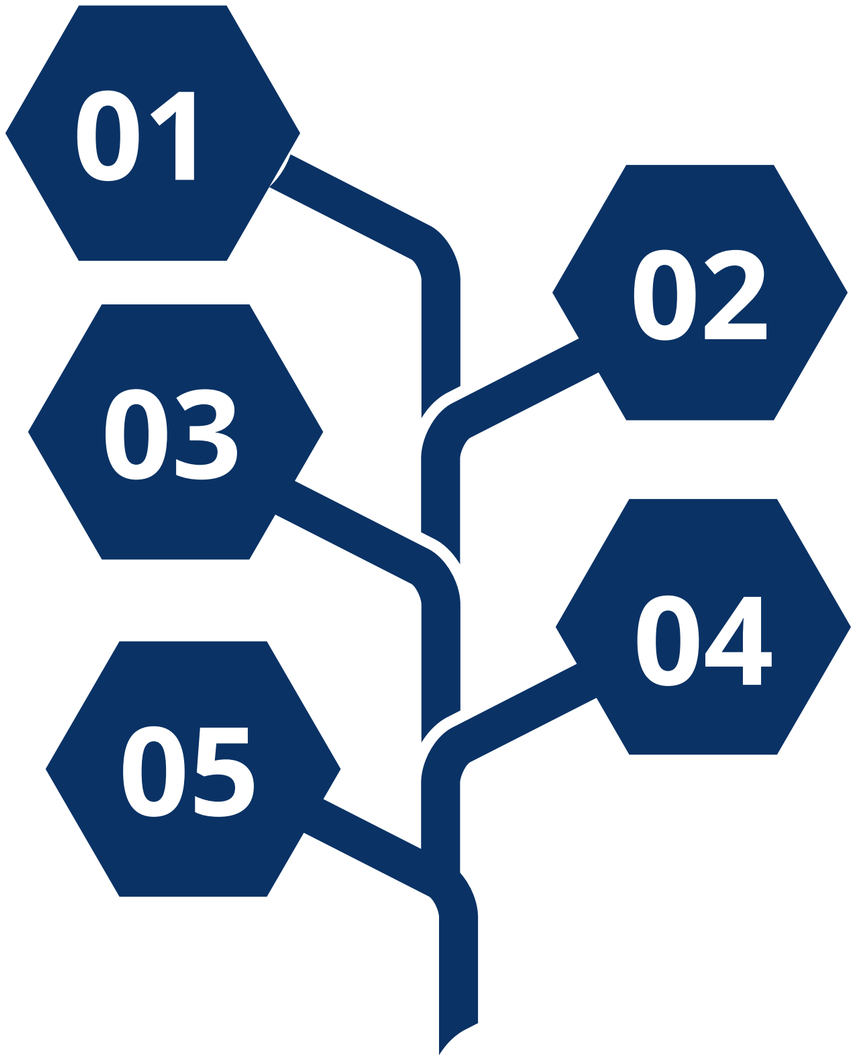
Completed by : Meriem BRAHEM & Salsabil JABALAH

Supervised by : Mr. Samir SAKRANI

3AGE2

INTRODUCTION

Table of
Contents



Features about ATmega328P

High performance, low power AVR® 8-bit microcontroller
Advanced RISC architecture

- **32 * 8 general purpose working registers**
- **Up to 16MIPS throughput at 16MHz**

High endurance non-volatile memory segments

- **32K bytes of in-system self-programmable flash program memory**
- **1Kbytes EEPROM**
- **2Kbytes internal SRAM**

Peripheral features

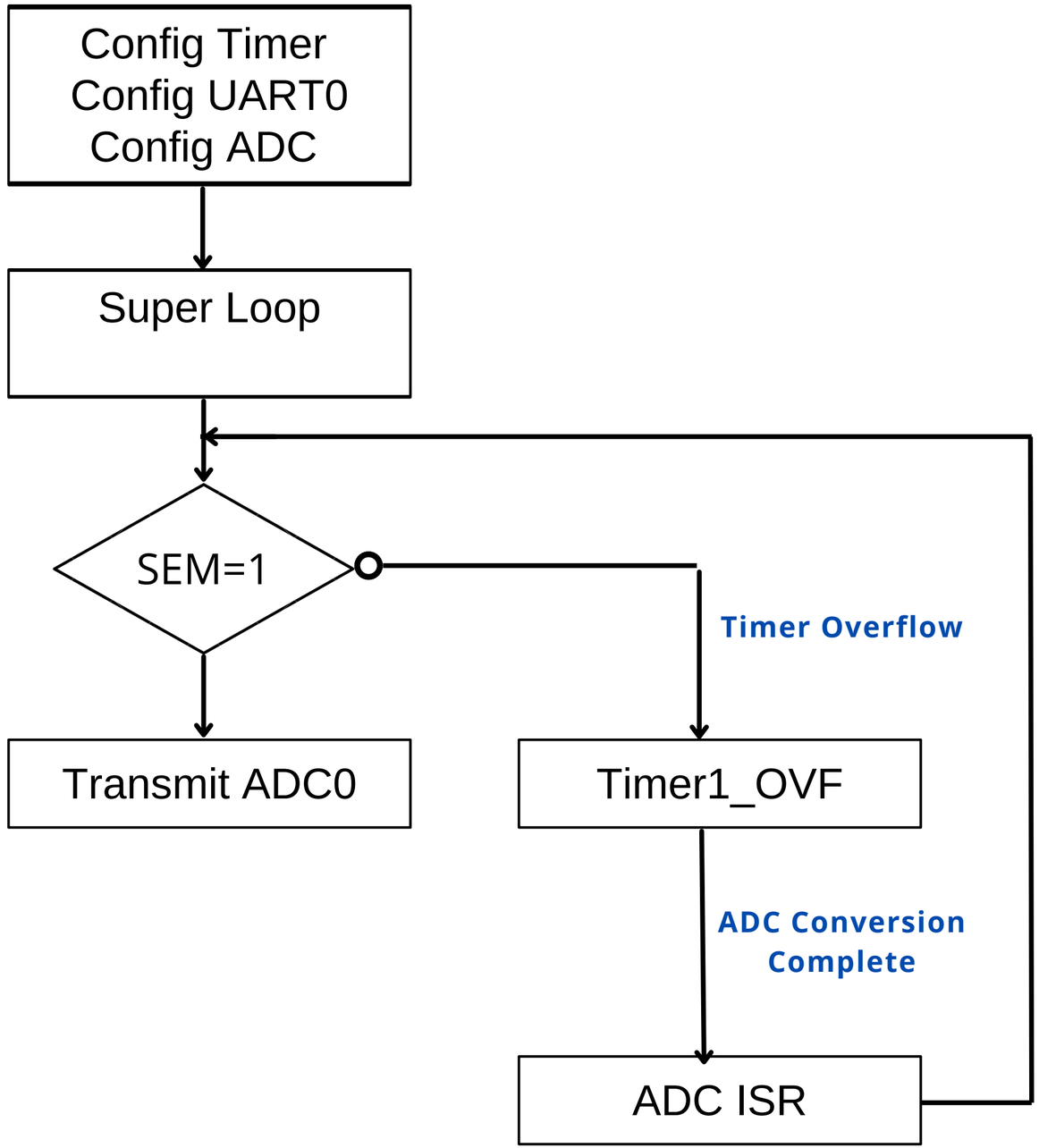
- **Two 8-bit Timer/Counters with separate prescaler and compare mode**
- **One 16-bit Timer/Counter with separate prescaler, compare mode, and capture mode**
- **Six PWM channels**
- **8-channel 10-bit ADC in TQFP and QFN/MLF package**
- **Programmable serial USART**
- **Master/slave SPI serial interface**
- **Byte-oriented 2-wire serial interface (Phillips I2C compatible)**

Project Overview:

This project is based on the ATmega328P microcontroller and demonstrates the integration of three main components: ADC (Analog-to-Digital Conversion), UART communication, and Timer Interrupts. It aims to continuously monitor an analog input (using the ADC) and send the corresponding digital value over UART when triggered by a timer overflow interrupt every 500 ms . Below is a breakdown of the key aspects:

config_timer: Configures the Timer1 in normal mode to trigger overflow interrupts.
config_uart0: Configures UART0 for communication at 9600 bps.
config_adc: Configures the ADC to read from channel 0 .
trans_data: Sends a single byte of data over UART**.**
trans_string: Sends a string of characters over UART**.**
trans_adc7: Formats and sends the ADC result over UART**.**
TIMER1_OVF: Timer overflow interrupt handler that toggles an LED and triggers an ADC conversion.
ADC: ADC interrupt handler that sets the semaphore to signal when the data is ready for transmission.

Chronological Flow of the Program



The universal synchronous and asynchronous serial receiver and transmitter USART0

Introduction

- USART is a hardware communication protocol used for serial communication between devices. It allows data to be transmitted and received one bit at a time over a single data line, making it ideal for low-cost setups.

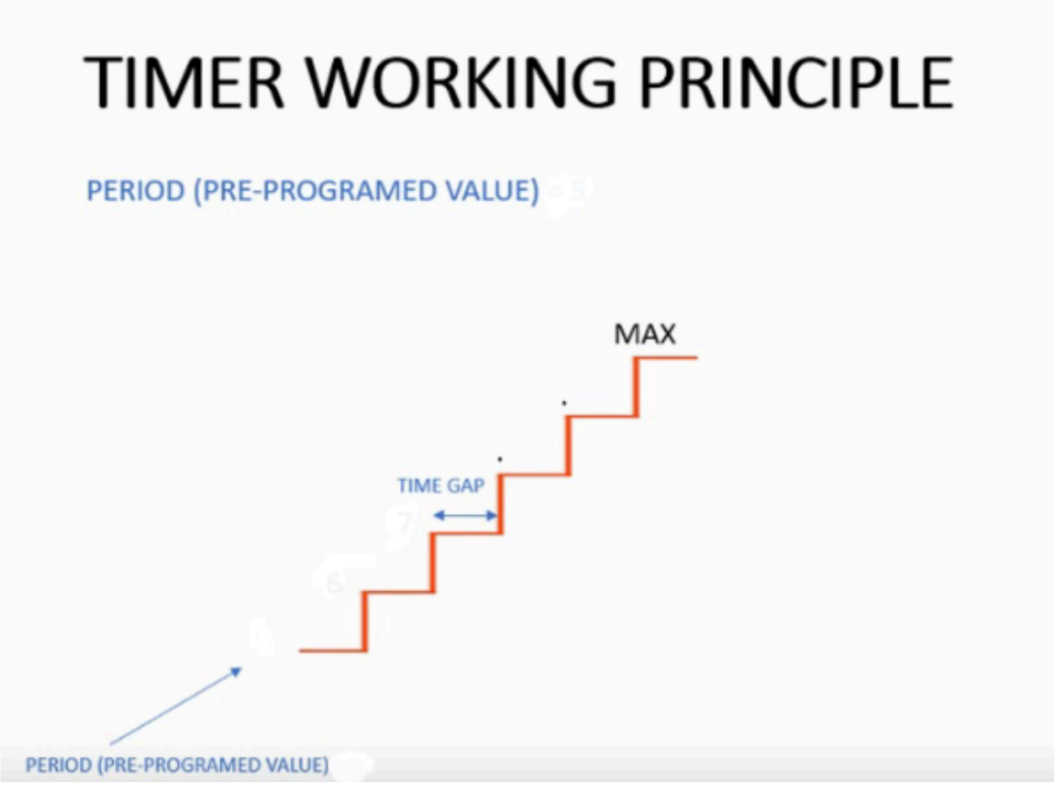
Register Configuration

- Enables both transmitter (TX) and receiver (RX)
- Configures the frame format with 8 data bits, 1 stop bit, and no parity
- This configuration sets up UART0 for asynchronous communication
- Enables double-speed mode (U2X0) for high-speed communication
- Sets the baud rate to 9600 bps

16-bit Timer/Counter1
TIMER INTERRUPT MODE IN AVR MICROCONTROLLER

Introduction

The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement.



Register Configuration

- Enable the Global interrupt interrupt bit
- Enable the required Timer Interrupt : Overflow Interrupt Enable
- Set the Timer Mode : Mode normal
- Set the prescaler for the Timer : clk/1024 (from prescaler)
- Set the Period for the Timer(Pre-Programed value):
- The system clock is 8 Mhz
- The Clock frequency becomes : 8Mhz/64 =125kHz
- Time for each Tick = 1/125000 = 8 us
- For 500 ms to elapase the count is 500ms/ 8 us = 62500
- So 65535 - 62500 = 3035

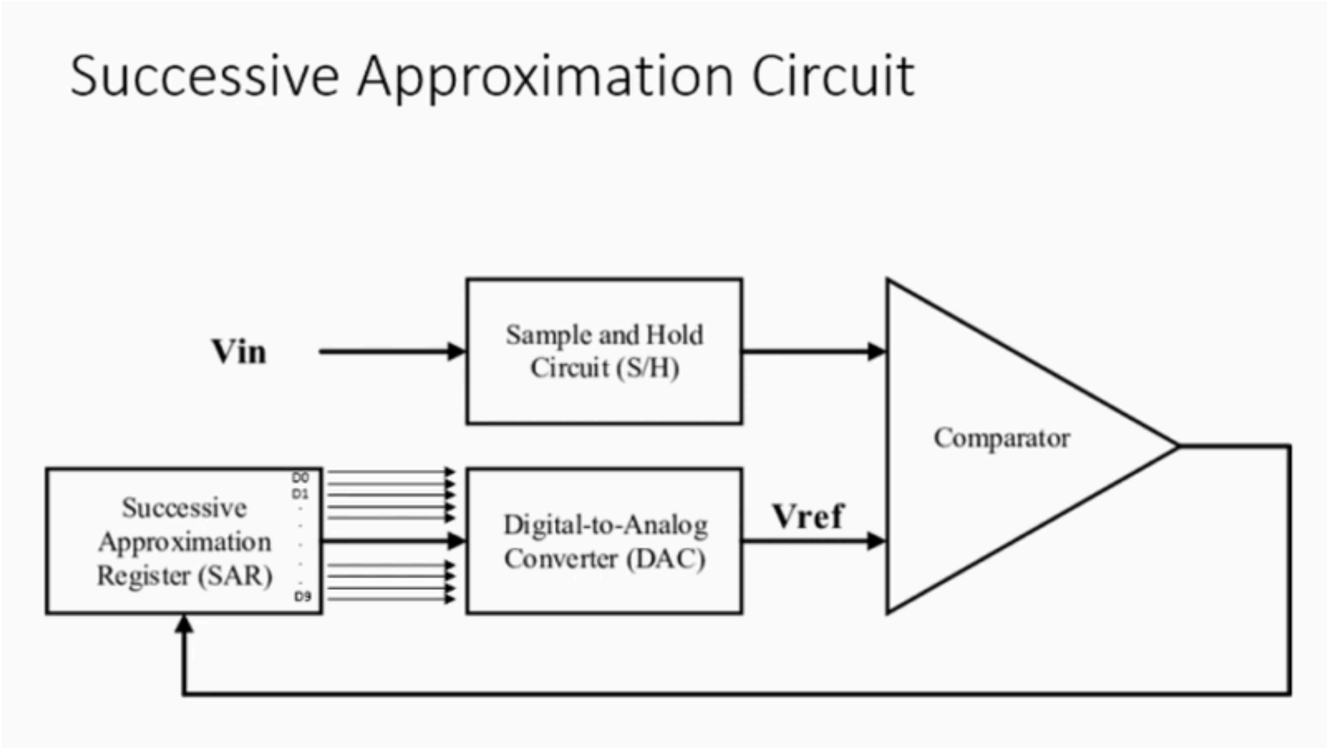
Implement the Interrupt Handler Function : TIMER1_OVF

- Modifying PORTB : PB7 (LED Toggle)
- Reloading the Timer Counter Register
- Starting an ADC Conversion :write ADSC bit to one to start each conversion

Analog-to-Digital Converter

Introduction

- **10-bit resolution**
- **65 to 260µs conversion time**



- The ADC is connected to an 8-channel analog multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).

- The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin .

Register Configuration

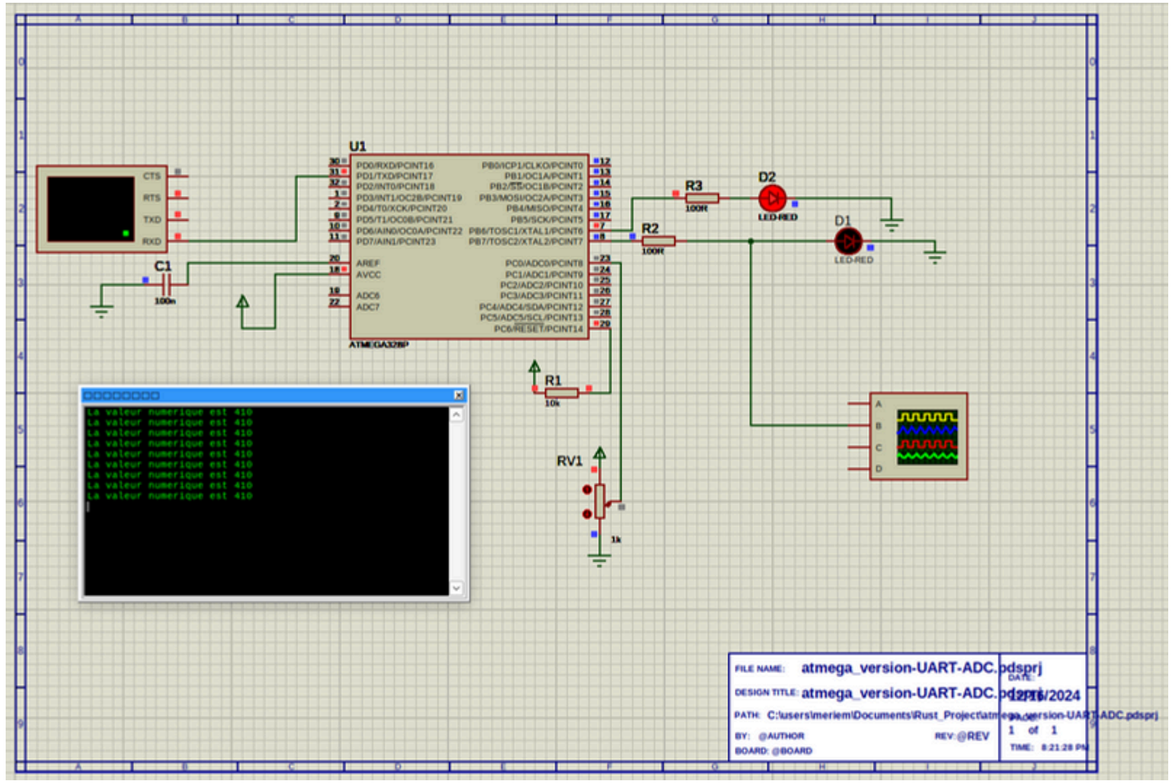
- Enable the global Interrupt bit
- Set voltage reference selections for ADC :AVCC with external capacitor at AREF pin
- Set input : ADC0 Channel
- No auto-triggering is configured : Auto-triggering of the ADC is disabled
- Set data alignment in data register :The ADC data is right-adjusted .
- Enable ADC and set Prescaler = 8

Implement the Interrupt Handler Function :ADC

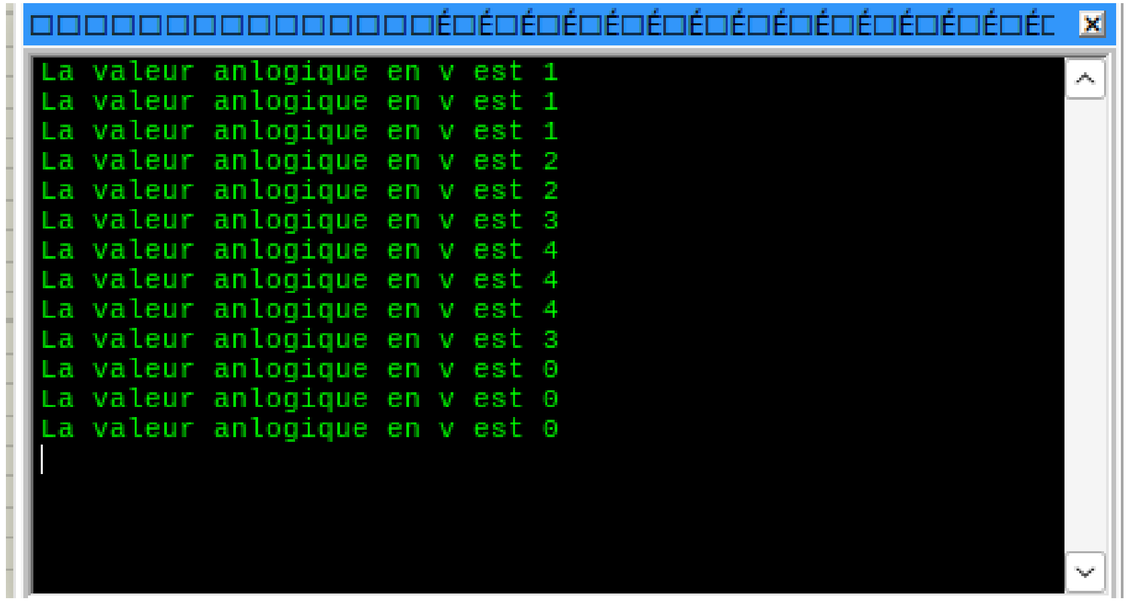
- This ISR is called whenever the ADC conversion is complete
- The semaphore (SEM - an atomic boolean) is set to true to trigger a part in the super loop to start sending the data that was collected during the ADC conversion.

Resultat de conversion d’une valeur analogique de potentiometre :

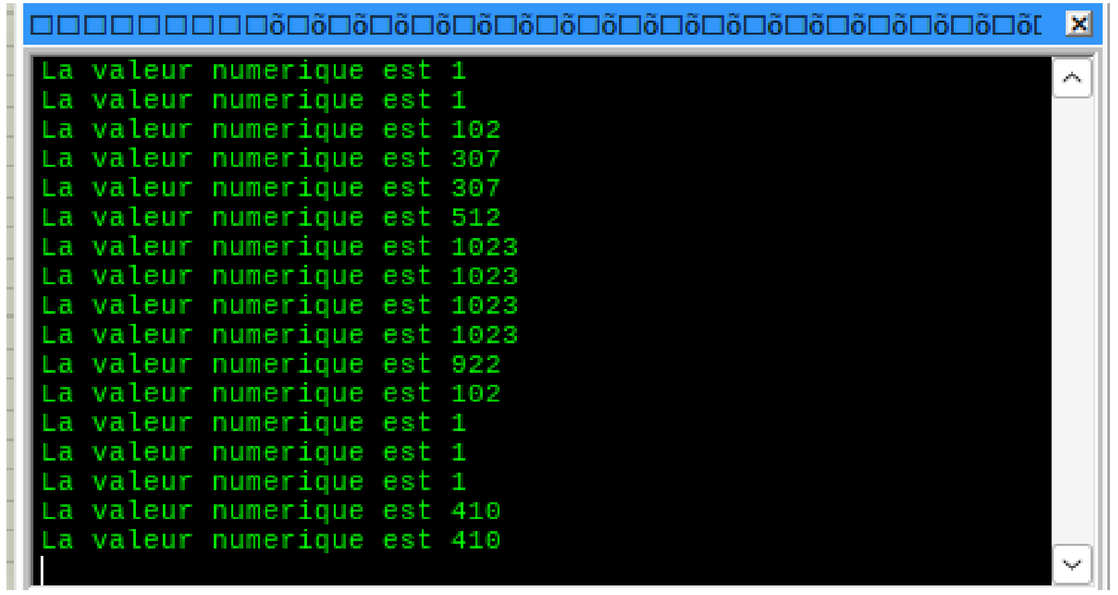
- UART Conversion and Transmission: The conversion of the ADC integer value to a string using itoa is well-managed for UART transmission every T=500 ms + ADC Conversion Time :
- Led D1 :PB 7 toggle every 500 ms
- Led D2 :PB 6 toggle every 500 ms + ADC Conversion Time



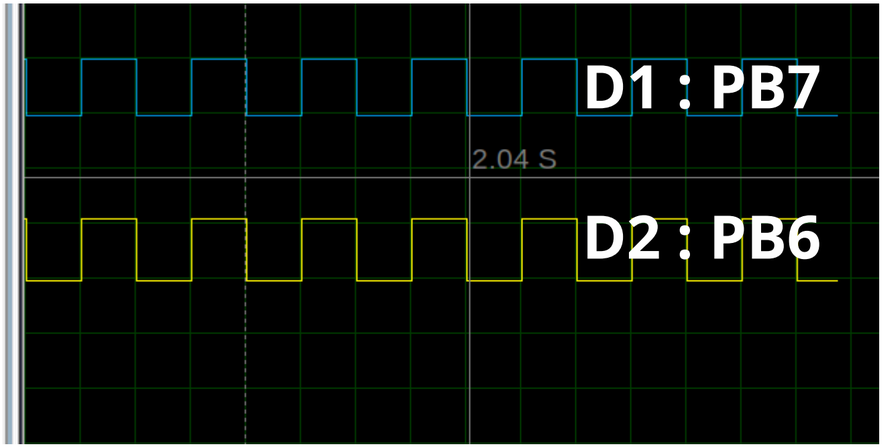
Proteus simulation



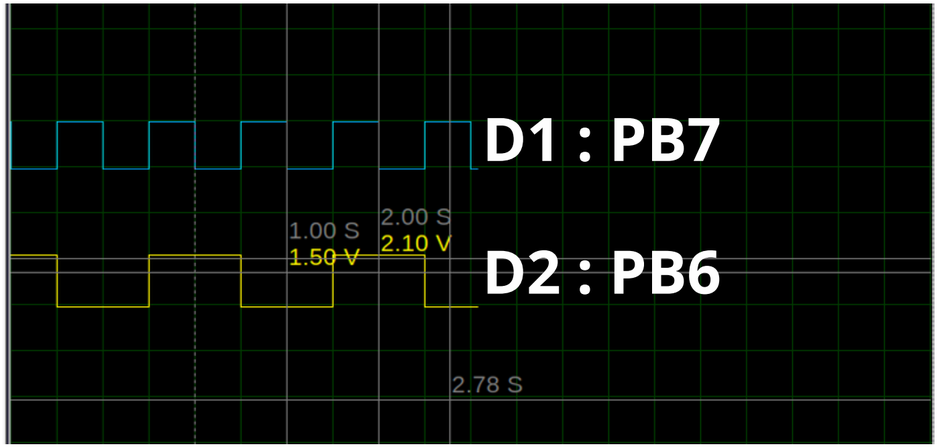
Virtual terminal output



Virtual terminal output



Digital oscilloscope output without uart transmission



Digital oscilloscope with uart transmission

Interpretation

Digital oscilloscope output with uart transmission:

- The UART blocks the execution of the main loop while it is transmitting data, which can delay or temporarily prevent tasks such as toggling PB6. Although the timer interrupt and ADC interrupt are executed immediately when triggered, the main loop is stalled due to the UART's blocking behavior, leading to a delay in toggling PB6.

Digital oscilloscope output without uart transmission:

- The short delay due to the ADC conversion time is not visible when UART is not active because the conversion time is so small (typically in the microsecond range).
- The synchronization between PB6 and PB7 is expected without UART, as the delays from the ADC conversion are minimal and often not visible.

Bibliography

ATmega328P Datasheet

Register Level Programming in Arduino :https://www.youtube.com/watch?v=kVVgsbXYp1I&list=PL_zvrXFdKgZqiTxipn7WjT_T9hU7ughQH&index=4

Conclusion

Thank You!