



Mini Oscilloscope

The potentiometer reads the desired value through ADC, then is given to the timer to generate a PWM signal using Phase Correct PWM, and then displayed on the graphical LCD through I2C Protocol.

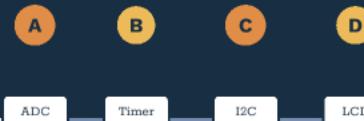
Project

PWM drawer works as a mini oscilloscope that is responsible for printing the frequency and the waveform of the generated PWM signal.

? Why?
measure, debug

? How?

Milestones



Hardware Modules





Project

PWM drawer works as a mini oscilloscope that is responsible for printing the frequency and the waveform of the generated PWM signal.



Why?

measure, debug



How?

Milestones

A

ADC

B

Timer

C

I2C

D

LCD



Hardware Modules

1

ATMega328p

2

SSD1306

3

Potentiometer

1 ATMega328p

- 8-bit Architecture
- Harvard Architecture
- 32 KB of Flash memory
- 2 KB of SRAM
- 1 KB of EEPROM
- Various timers/ counters, PWM, USART, SPI, I2C, ADC, etc.
- Low Power Consumption
- Cost-effective and widely available
- Strong Community Support

Pins

- Input: ADC: PC3
- Communication: SDA: PC4, SCL: PC5
- Output: PWM: PD6

ATmega328

Pin-out
Figure 5-1. 28-pin PDIP

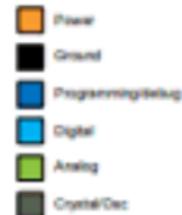
(PCINT14/RESET) PC6	1	28	PCS (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

ATmega328

Pin-out

Figure 5-1. 28-pin PDIP

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)



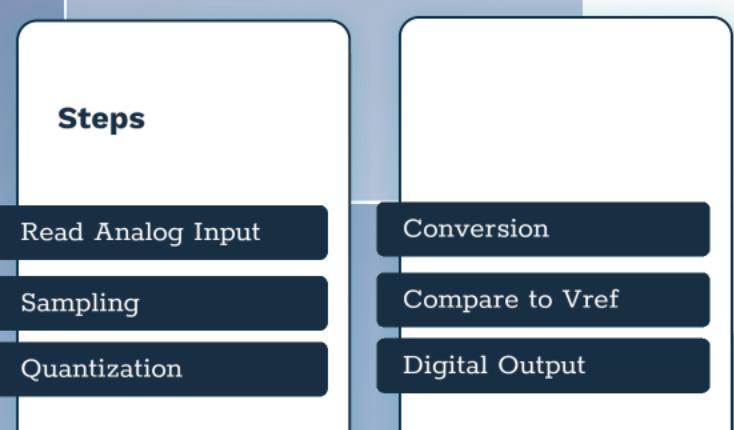
Pins

- Input: ADC: PC3
- Communication: SDA: PC4, SCL: PC5
- Output: PWM: PD6

2 ADC

ADC.h -> ADC.c

- 2 Methods: void adc_init(); uint16_t adc_read();
- Set reference voltage to AVCC, and select ADC3 as the default input channel
- Enable ADC, and set the prescaler to 128 (for example)
- Start the conversion
- Wait for the conversion to complete, Return the ADC result



In electronics, an analog-to-digital converter is a system that converts an analog signal, such as a sound picked up by a microphone or light entering a digital camera, into a digital signal.

Potentiometer

Steps

Read Analog Input

Sampling

Quantization

Conversion

Compare to Vref

Digital Output

ADC.h -> ADC.c

- 2 Methods: `void adc_init(); uint16_t adc_read();`
- Set reference voltage to AVCC, and select ADC3 as the default input channel
- Enable ADC, and set the prescaler to 128 (for example)
- Start the conversion
- Wait for the conversion to complete, Return the ADC result

3 Timers: PWM

- Phase Correct PWM generates a symmetric waveform
- It improves motor control
- It provides smoother and more precise control over the motor speed

Timer.h -> Timer.c

- 2 Methods: void initPWM();
void setPWMDutyCycle(uint16_t value);
- Set OC0A (Pin 6, PD6) as output
- Set prescaler to 8
- Enable Phase Correct PWM mode, non-inverting output on OC0A
- Calculate the corresponding PWM value for the given ADC value
- Set the OCR0A register with the calculated PWM value

Steps

Select Pin

Set prescaler

PWM mode

Non-inverting mode

Input ADC: 0 to 1024,
Output OCROA
Prescaler(8): 0 to 255/8
 $\rightarrow \text{pwmValue} = (255.0 / 1024 * 8.0) * \text{value};$

Steps

Read Ana

Sampling

Quantizat

Steps

Select Pin

Set prescaler

PWM mode

Non-inverting mode

Input ADC: 0 to 1024,
Output OCROA
Prescaler(8): 0 to 255/8
 $\rightarrow \text{pwmValue} = (255.0 / 1024 * 8.0) * \text{value};$

Timer.h -> Timer.c

- 2 Methods: `void initPWM();`
- ```
void setPWMDutyCycle(uint16_t value);
```
- Set OC0A (Pin 6, PD6) as output
  - Set prescaler to 8
  - Enable Phase Correct PWM mode, non-inverting output on OC0A
  - Calculate the corresponding PWM value for the given ADC value
  - Set the OCR0A register with the calculated PWM value



# 3 Graphical LCD

SSD1306



## Steps

Setup SSD1306 Driver

Initialize I2C Address

Initialize Screen interface

Set screen font

Draw on screen text

Draw on screen square  
wave

## Draw Methods

- String: DrawStr(i2c adress, x\_pos, y\_pos, text\_buffer[]);
- Draw Line: DrawLine(i2c address, x\_init\_pos, y\_init pos, x\_end\_pos, y\_end\_pos);

3 -

- Phase Corr  
a symmetric
- It improves
- It provides  
precise con  
speed

## Steps

Setup SSD1306 Driver

Initialize I2C Address

Initialize Screen interface

Set screen font

Draw on screen text

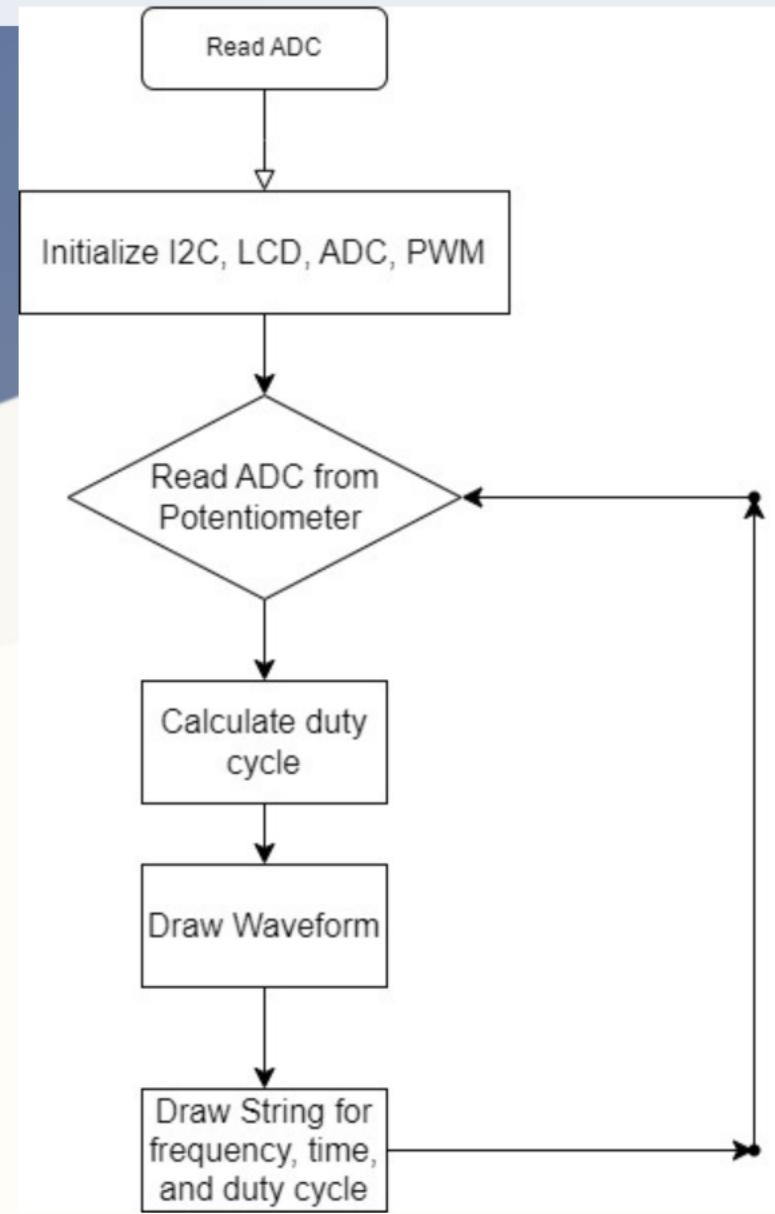
Draw on screen square  
wave

# Draw Methods

---

- String: DrawStr(i2c adress, x\_pos, y\_pos, text\_buffer[]);
- Draw Line: DrawLine(i2c address, x\_init\_pos, y\_init pos, x\_end\_pos, y\_end\_pos);

# main.c



$$DutyCycle_{normal} = \frac{OCR0A}{255} \cdot 100$$

Output

1  
2  
3  
4  
SCL  
SDA

LCD2  
OLED12864I2C

Control Section

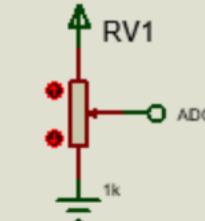
U1

|    |                       |
|----|-----------------------|
| 30 | PD0/RXD/PCINT16       |
| 31 | PD1/TXD/PCINT17       |
| 32 | PD2/INT0/PCINT18      |
| 1  | PD3/INT1/OC2B/PCINT19 |
| 2  | PD4/T0/XCK/PCINT20    |
| 9  | PD5/T1/OC0B/PCINT21   |
| 10 | PD6/AIN0/OC0A/PCINT22 |

Output



Input



## Control Section

U1

|    |                       |    |                        |
|----|-----------------------|----|------------------------|
| 30 | PD0/RXD/PCINT16       | 12 | PB0/ICP1/CLK0/PCINT0   |
| 31 | PD1/TXD/PCINT17       | 13 | PB1/OC1A/PCINT1        |
| 32 | PD2/INT0/PCINT18      | 14 | PB2/SS/OC1B/PCINT2     |
| 1  | PD3/INT1/OC2B/PCINT19 | 15 | PB3/MOSI/OC2A/PCINT3   |
| 2  | PD4/T0/XCK/PCINT20    | 16 | PB4/MISO/PCINT4        |
| 9  | PD5/T1/OC0B/PCINT21   | 17 | PB5/SCK/PCINT5         |
| 10 | PD6/AIN0/OC0A/PCINT22 | 18 | PB6/TOSC1/XTAL1/PCINT6 |
| 11 | PD7/AIN1/PCINT23      | 19 | PB7/TOSC2/XTAL2/PCINT7 |
| 20 | AREF                  | 23 | PC0/ADC0/PCINT8        |
| 18 | AVCC                  | 24 | PC1/ADC1/PCINT9        |
| 19 | ADC6                  | 25 | PC2/ADC2/PCINT10       |
| 22 | ADC7                  | 26 | PC3/ADC3/PCINT11       |
|    |                       | 27 | PC4/ADC4/SDA/PCINT12   |
|    |                       | 28 | PC5/ADC5/SCL/PCINT13   |
|    |                       | 29 | PC6/RESET/PCINT14      |

ATMEGA328P

