

AMIT GRADUATION PROJECT

PWM Drawer

PRESENTED BY:

MOHAMED HASSAN ALI

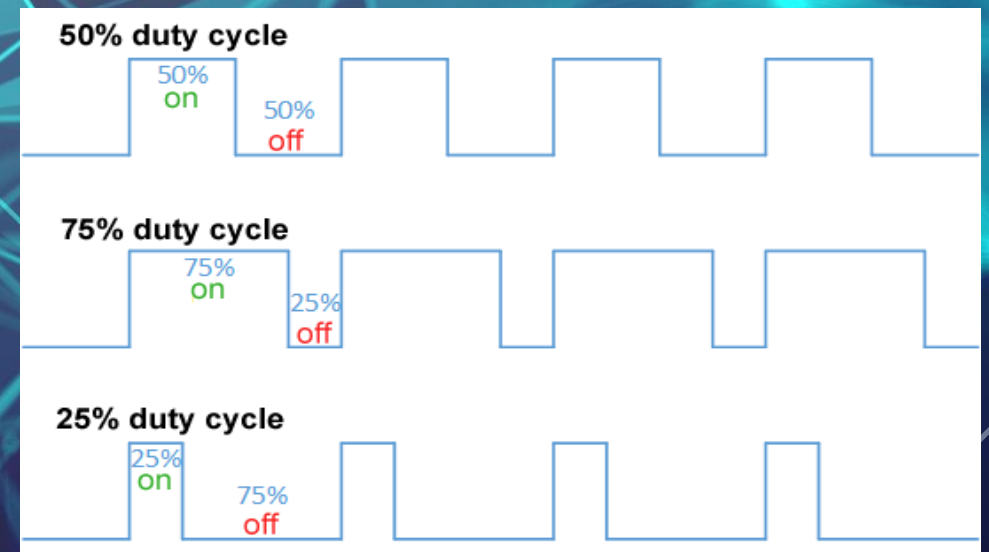
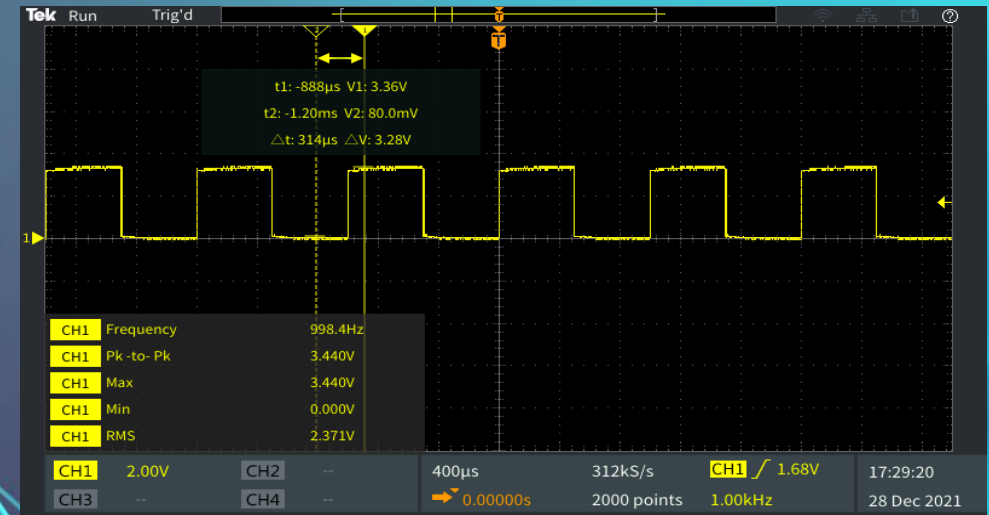


TABLE OF CONTENTS:

1. Introduction
2. Components
3. Layered Architecture
4. Flow Chart
5. Special Pattern
6. Final Project

1. Introduction

- Pulse Width Modulation is a technique used to control analog devices, using a digital signal. This technique can be used to output an analog-like signal from a digital device, like a microcontroller. We can control motors, lights, actuators, and more using the generated PWM signal. An important thing to note here is that PWM is not a true analog signal. The digital signal is modified in a way to fake an analog signal.



Aim of The Project

- This Project Aims to Use ATmega32 Microcontroller's Input Capture Unit to Measure PWM Signal That Comes from other sources or the Microcontroller itself.

2. Components

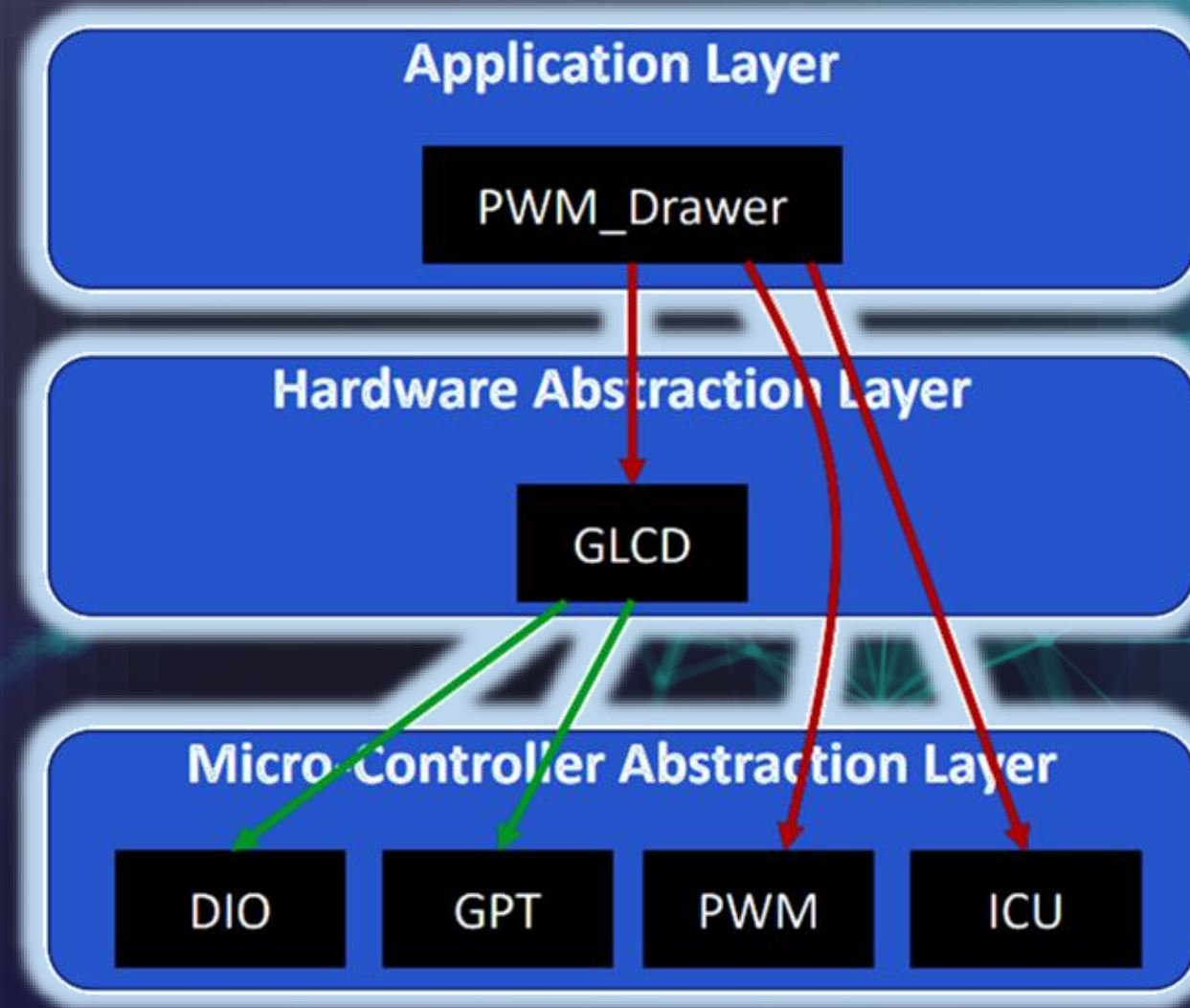


128 X 64 Graphical LCD

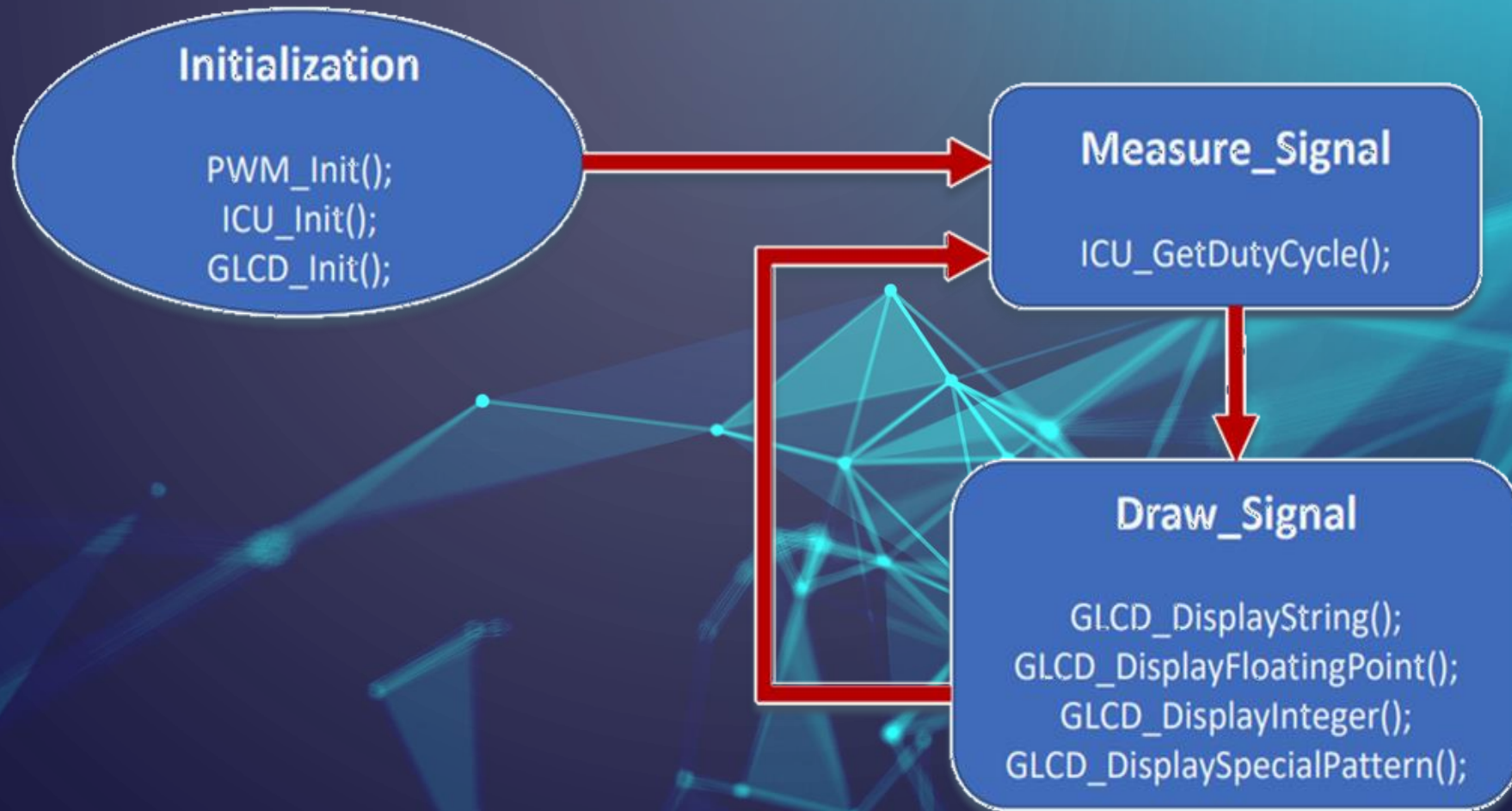
				ATmega 16/32			
(XCK/T0)	PB0	1		40	PA0	(ADC0)	
(T1)	PB1	2		39	PA1	(ADC1)	
(INT2/AIN0)	PB2	3		38	PA2	(ADC2)	
(OC0/AIN1)	PB3	4		37	PA3	(ADC3)	
(SS)	PB4	5		36	PA4	(ADC4)	
(MOSI)	PB5	6		35	PA5	(ADC5)	
(MISO)	PB6	7		34	PA6	(ADC6)	
(SCK)	PB7	8		33	PA7	(ADC7)	
RESET		9		32	AREF		
VCC		10		31	AGND		
GND		11		30	AVCC		
XTAL2		12		29	PC7	(TOCS2)	
XTAL1		13		28	PC6	(TOCS1)	
(RXD)	PD0	14		27	PC5	(TD1)	
(TXD)	PD1	15		26	PC4	(TD0)	
(INT0)	PD2	16		25	PC3	(TMS)	
(INT1)	PD3	17		24	PC2	(TCK)	
(OC1B)	PD4	18		23	PC1	(SDA)	
(OC1A)	PD5	19		22	PC0	(SCL)	
(ICP1)	PD6	20		21	PD7	(OC2)	

ATMega32

3. *Layered Architecture*



4. Flow Chart (A)



4. Flow Chart (B)

Measure_Signal

ICU_GetDutyCycle();

Clear Input Capture Flag
Set Trigger Edge: RISING_EDGE

Wait for Input Capture → i.e. Start of Cycle

Clear Input Capture Flag and Timer Counter
Set Trigger Edge: FALLING_EDGE

Wait for Input Capture → i.e. High Count

Clear Input Capture Flag and Timer Counter
Set Trigger Edge: RISING_EDGE

Wait for Input Capture → i.e. Low Count

$$\text{Duty Cycle \%} = \frac{\text{High Time}}{\text{Period Time}} \times 100$$

$$\text{High Time ms} = \frac{\text{High Count} * \text{Prescaler} * 10^3}{F_{\text{CPU}}}$$

$$\text{Period Time ms} = \frac{(\text{High Count} + \text{Low Count}) * \text{Prescaler} * 10^3}{F_{\text{CPU}}}$$

4. Flow Chart (C)

Draw_Signal

```
GLCD_DisplayString();  
GLCD_DisplayFloatingPoint();  
GLCD_DisplayInteger();  
GLCD_DisplaySpecialPattern();
```

Choose a scale (milliseconds
to pixel) as $\frac{\text{Period Time}}{5}$.

GLCD Line 0: Display Frequency Value in kHz.

GLCD Line 1: Display Duty Cycle Value in %.

GLCD Line 4: Display Period Time Value in
milliseconds.

GLCD Line 5: Display Arrow on First Cycle
Period Time.

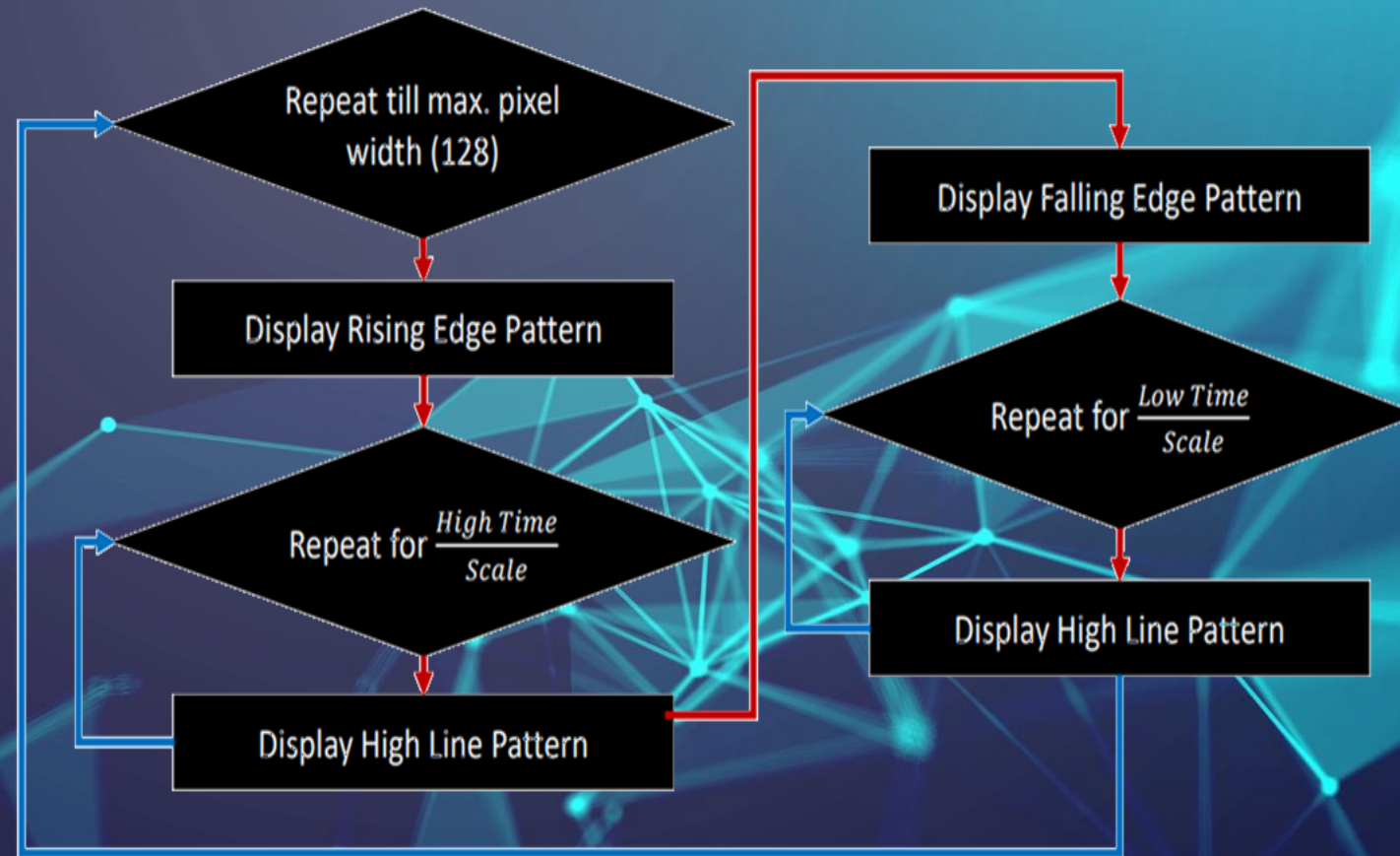
GLCD Line 6: Display the PWM signal shape.

4. Flow Chart (D)

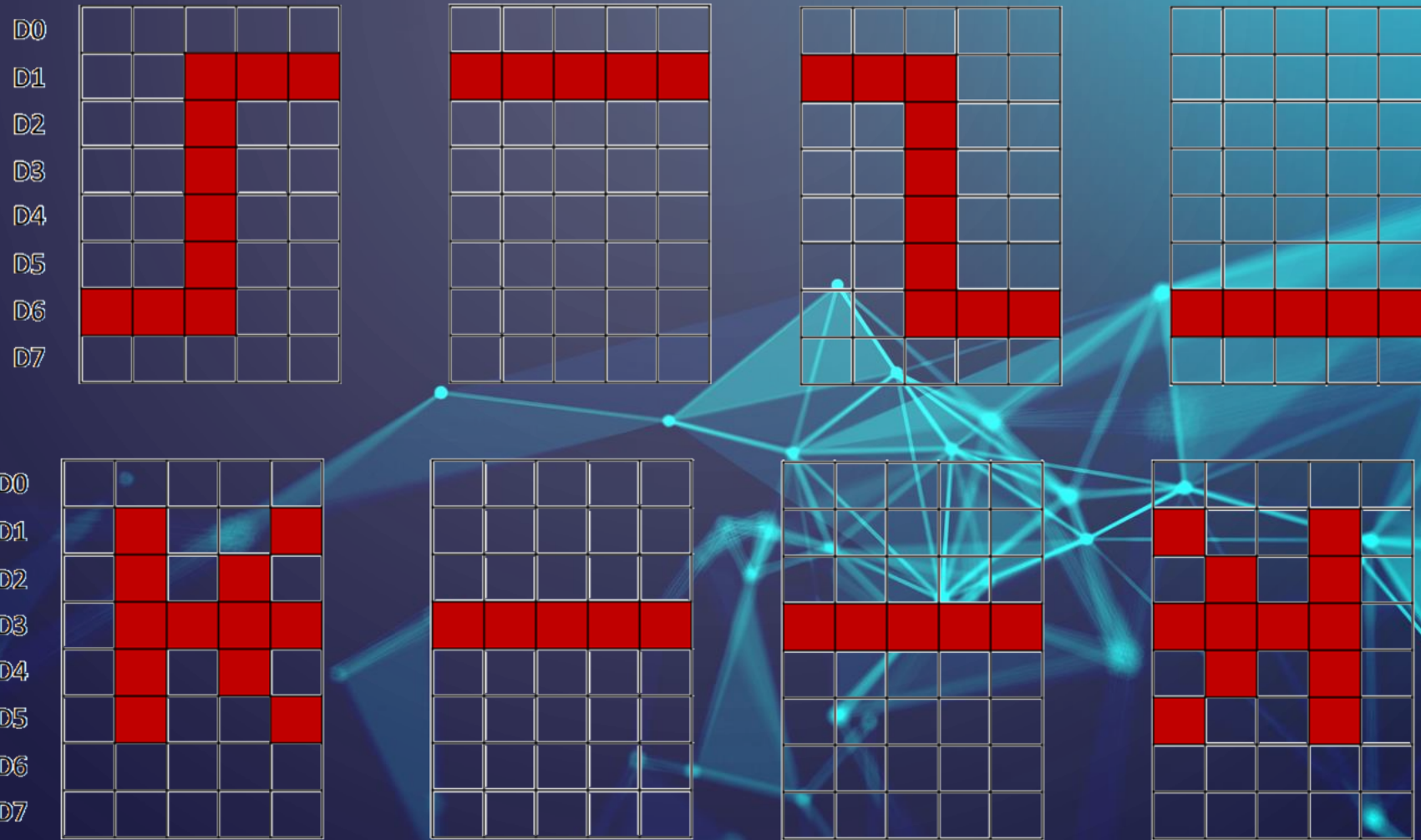
Draw_Signal

```
GLCD_DisplayString();  
GLCD_DisplayFloatingPoint();  
GLCD_DisplayInteger();  
GLCD_DisplaySpecialPattern();
```

GLCD Line 6: Display the PWM
signal shape.



5. *Special Pattern*

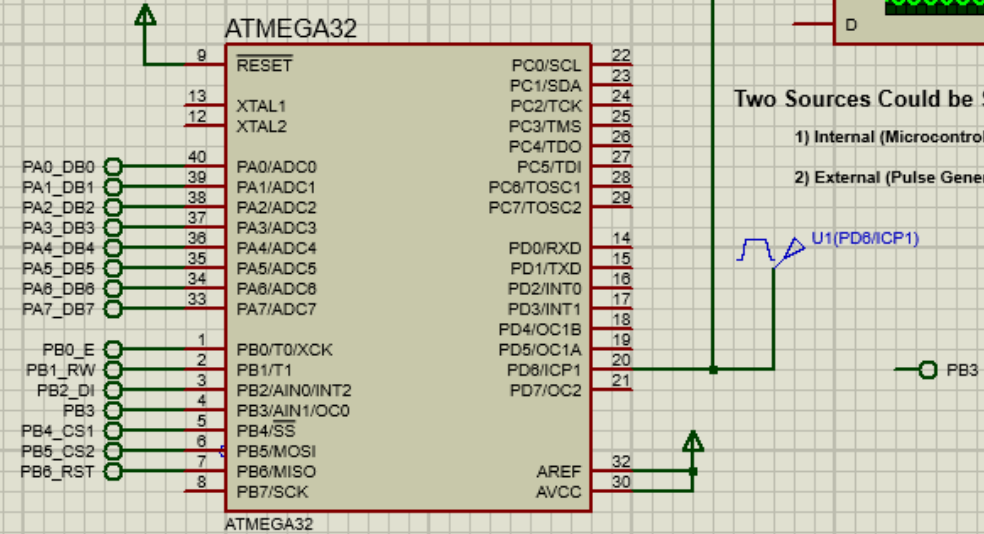
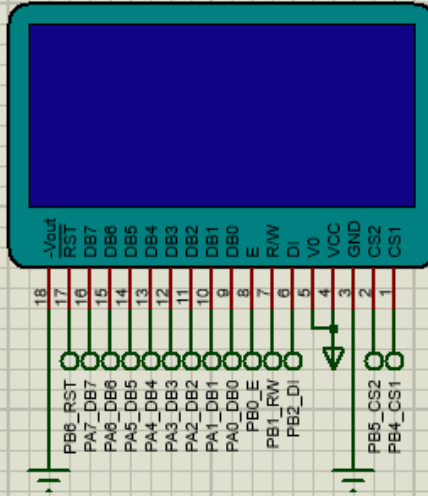


6. Final Project

Mohamed Hassan Ali Hassan
AMIT Graduation Project
PWM Drawer

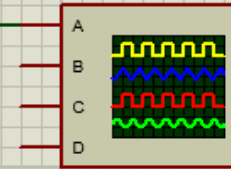
Note: Make Sure ATmega32 Frequency is: 16 MHz

GLCD
LGM12841BS1R



Two Sources Could be Selected for PWM:

- 1) Internal (Microcontroller itself)
- 2) External (Pulse Generator or Multivibrator)



The background is a dark blue gradient. It features white circuit-like lines with circular nodes at the corners. A prominent cyan-colored network graph, composed of interconnected nodes and lines, is visible on the right side. The text is centered in a white, italicized serif font.

Thank You
Any Questions?