[redacted]

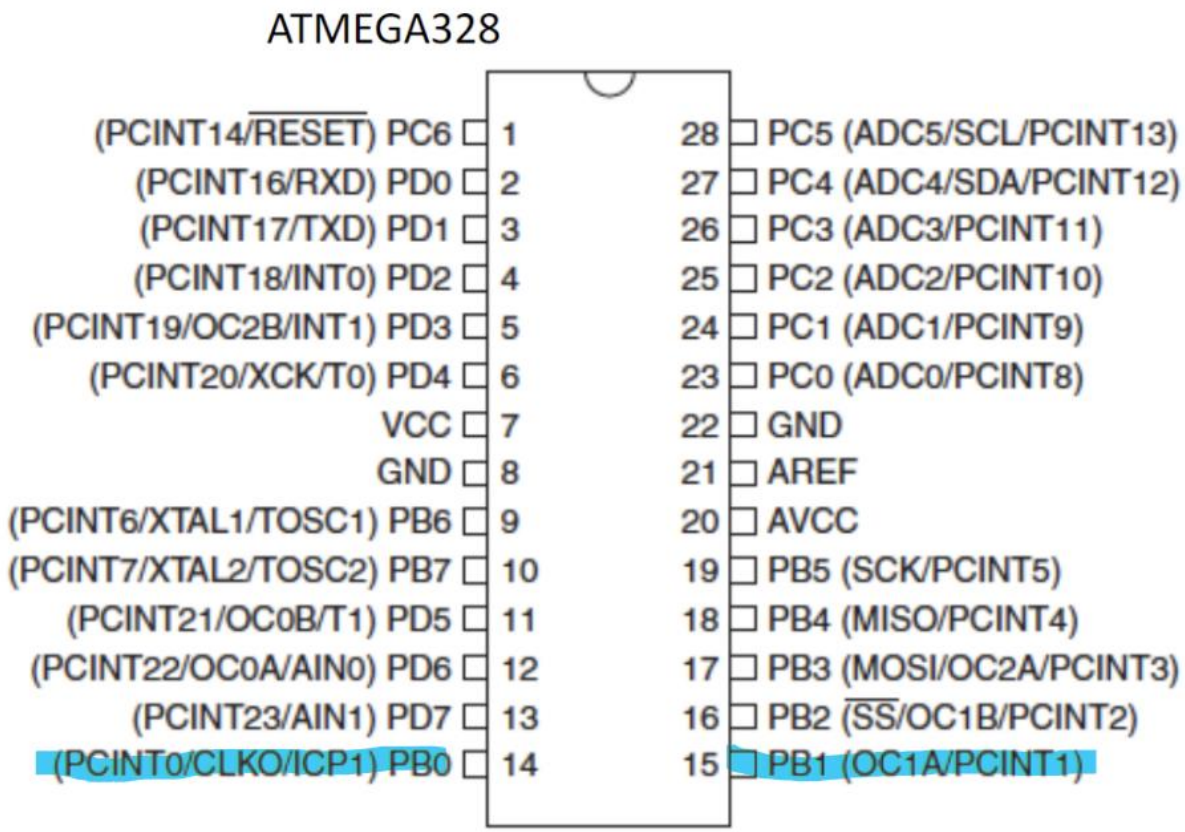
04 Distance Sensor Radar

Student Name: [redacted]
Student #: [redacted]
Student Email: [redacted]
Primary Github address: https://github.com/DylanCaz/Submission_DA.git
Directory:
https://github.com/DylanCaz/Submission_DA/tree/main/Design_Assignments_sub/DA_4_sub

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS



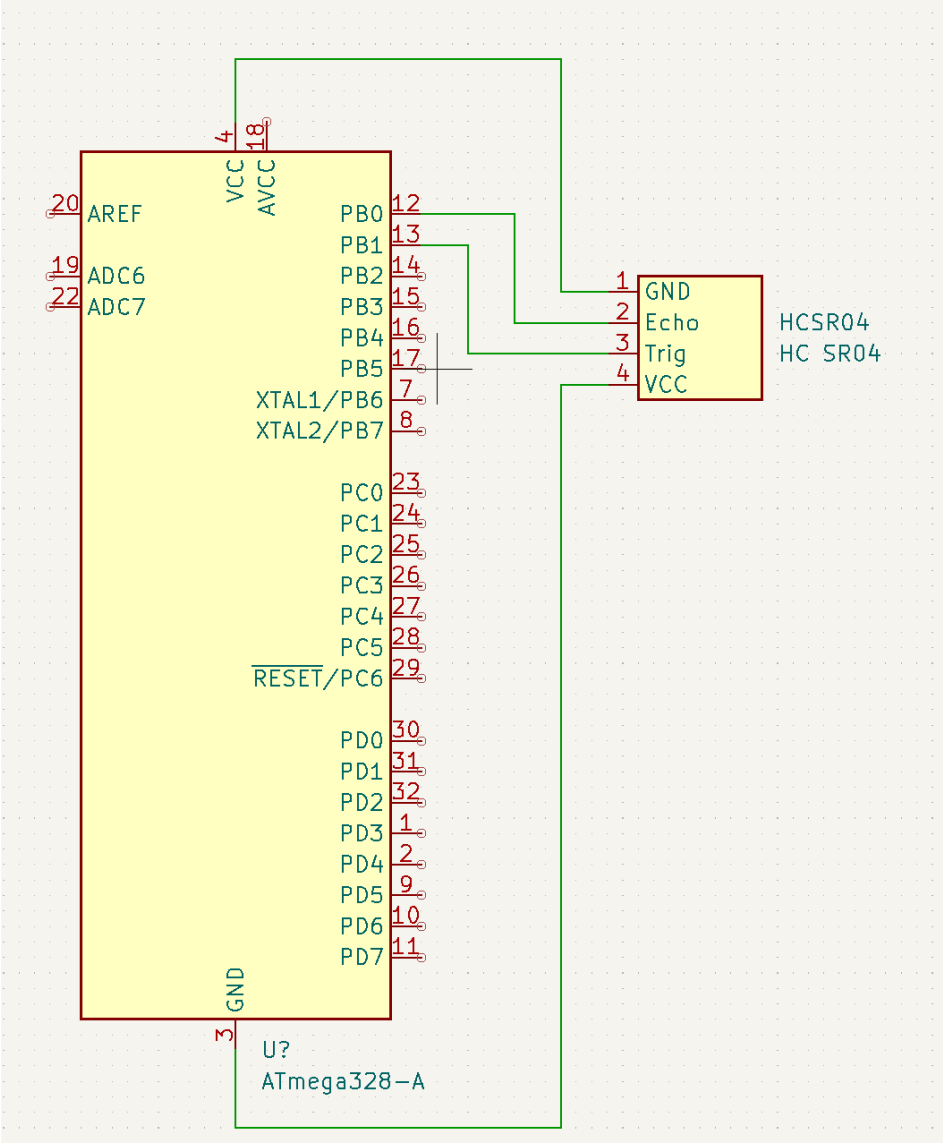
2. DEVELOPED MODIFIED CODE OF TASK 1/2/3

```
#define F_CPU 16000000UL
#include <stdio.h>
#include <stdlib.h>
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <string.h>

#define BAUDRATE 9600
#define UBBR_9600 103 // 16Mhz with .2% error
#define Trigger_pin PB1 // Trigger Pin
int TimerOverflow = 0;

void USART_init(unsigned int ubrr); //Function to initialize and configure USART
void USART_send(unsigned char data); // Function to send char to serial port
void USART_tx_string( char *stringPtr ); // Function to send string to serial port

int main()
{
	char string[10];
	long cnt;
```

```
	double distance;

	DDRB = (1 << 0) | (1 << 1); // PB0 and PB1 as an output
	USART_init(UBBR_9600);

	sei(); // Global Interrupt
	TIMSK1 = (1 << TOIE1); // Enable Timer1 overflow interrupt
	TCCR1A = 0; // Setting bits to zero for Normal operation
	while(1)
	{
		PORTB |= (1 << Trigger_pin); // setting PB1 to trigger pin on HC-SR04
		_delay_us(340);
		PORTB &= ~(1 << Trigger_pin); // Toggle pin off

		TCNT1 = 0;
		TCCR1B = (1 << ICES1) | (1 < CS10); // No prescaler, Input Capture Noise
Canceler High
		TIFR1 = (1 << ICF1); // Clear Input Capture flag,
		TIFR1 =  (1 << TOV1); // Clear Timer Overflow flag

		while((TIFR1 & (1 << ICF1)) == 0); // Waiting for falling edge
		TCNT1 = 0; // Clear timer counter
		TCCR1B = (1 << CS10); // No prescaler, Input Capture Noise Canceler High
		TIFR1 = (1 << ICF1); // Clear Input Capture flag,
		TIFR1 =  (1 << TOV1); // Clear Timer Overflow flag
		TimerOverflow = 0; // Clear Timer overflow count

		while((TIFR1 & (1 << ICF1)) == 0); // Waiting for falling edge
		cnt = ICR1 + (65535 * TimerOverflow); // take count

		distance = (double)cnt/933; // 16Mhz Timer Frequency, sound speed = 343ms

		dtostrf(distance, 2, 2, string); // Convert distance into string
		strcat(string, " cm ");
		USART_tx_string("Distance = ");
		USART_tx_string(string);
		USART_tx_string("\r\n");
		_delay_ms(1000); // Task 3 waiting for 1 second after each output read

	}
}

//Function to initialize and configure USART
void USART_init(unsigned int ubrr)
{
	UBRR0H = (unsigned char)(ubrr>>8);
	UBRR0L = (unsigned char)ubrr;
	UCSR0B |= (1 << TXEN0) | (1 << RXEN0); // enable transmission and reception
	UCSR0B |= (1 << RXCIE0); // enable RX interrupt
	UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); // set frame formate to 8bits, no parity,
1
}
// Function to send char to serial port
void USART_send(unsigned char data)
{
	while (!(UCSR0A & (1 <<UDRE0)));
	UDR0 = data;
}
```

```
// Function to send string to serial port
void USART_tx_string(char* stringPtr)
{
	while ((*stringPtr != 0x00))
	{
		USART_send(*stringPtr);
		stringPtr++;
	}
}

ISR(TIMER1_OVF_vect)
{
	TimerOverflow++; // Increment timer overflow count
}
```

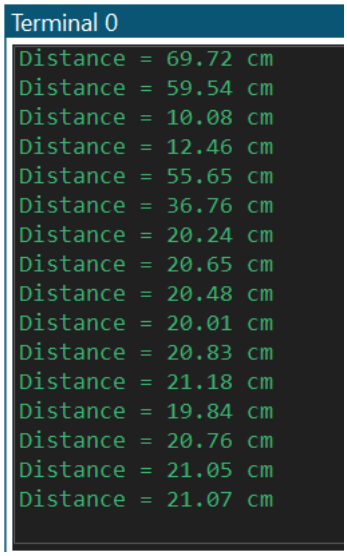
3. SCHEMATICS



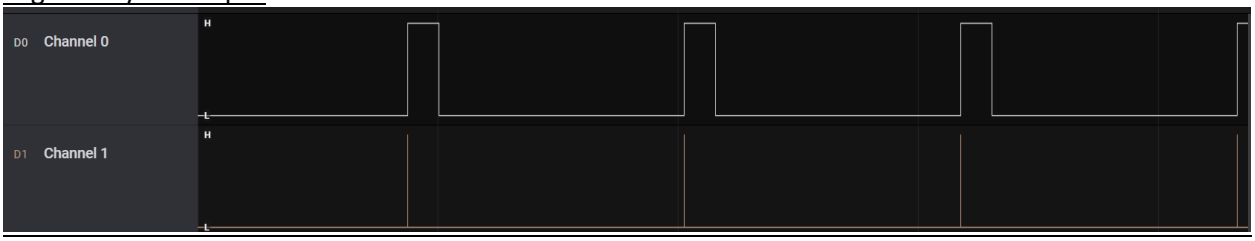
4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Successful Build

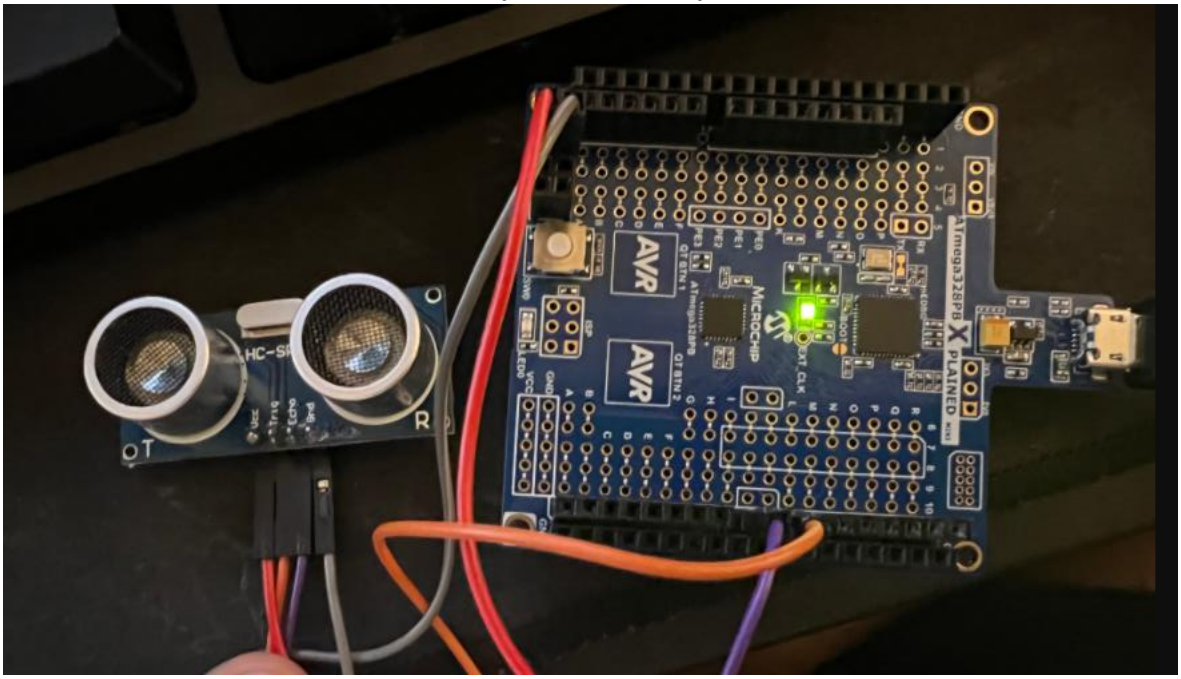
Terminal Window Output



Logic Analyzer Output



5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

https://youtu.be/6MvhiiGuJyE

7. GITHUB LINK OF THIS DA

https://github.com/DylanCaz/Submission_DA/tree/main/Design_Assignments_sub/DA_4_sub

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Dylan Cazares