## EE4524/ED5502 Spring 2024 Project 1 (Labs Weeks 5-8)

(Version 0.a, 15 Feb 2024, First draft version)

The aim of this project is to gain familiarity with programming a microcontroller, its timers and analog to digital converter using interrupts.

This project is worth **10%** of the total module assessment.

**General**

Write a single interrupt-driven program to do the following tasks concurrently:

1. Use the connected output port bits of an ATmega4809 microcontroller that are used on the Arduino Nano Every to implement a travelling single LED pattern (cylon eyes) on the LED shield provided with your Arduino board. The LED array is connected to the microcontroller’s output ports as shown in the Appendix. The on/off time of the LED must be set using a timer interrupt, where the on/off time is based on the input voltage applied to ADC input AIN3 of the microcontroller. You must **not** use a delay loop for on/off timing and you must write to the LED array in the Timer ISR.
2. Use the LED array to provide a graphical display of the input voltage applied to the ADC0 input 3 (AIN3) of the microcontroller in ‘thermometer’ format.
3. Read the input switch attached to PORTE bit 1, and use the value read to select between the use of LED array for a cylon eyes type display or the ADC value thermometer display. When the switch is NOT pressed (reads HIGH), display the cylon pattern; when the switch IS pressed (reads LOW), display the ADC value described in Point 2.
4. (Advanced Task) Use the ATmega4809 RTC timer/counter interrupt to switch between a ‘normal’ mode (displaying ‘1’ against a ‘0’s background), to an ‘inverse’ mode (displaying ‘0’ against a ‘1’s background), every 16 seconds. This only applies to the cylon display.

**In detail:**

PORTE bit 1 is connected to a push button switch on the shield. Use PORTE (read as PORTE.IN) input 1 to set a display mode on the LED output display.

If PORTE bit 1 is read as 1 (push button not pressed), the LED array is used for a ‘Cylon’ display.

In this case, the LED array displays the cylon pattern, where the on/off time is generated by the ATmega4809 Timer0 interrupt. You must NOT use delay loops to set the on/off time. The on/off time of the LED is to be based on the value read on ADC0 input AIN3. You must write to the LED array in the Timer Interrupt Service Routine.

For 0V <= ADC input 0 voltage <= 3V, the on/off time is 0.125 second

For 3V < ADC input 0 voltage <= 5V, the on/off time is 0.5 seconds

If PORTE bit 1 is read as 0 (push button pressed) the LED array is used to display an output indicating the ADC input voltage.

The output display uses the LED Array bits as follows:

|  |  |
| --- | --- |
| **Input Voltage Range** | **LED Array output display** |
| 0 <= ADC reading < 1/10 full scale | 00\_0000\_0001 |
| 1/10 <= ADC Reading < 1/5 full scale | 00\_0000\_0011 |
| 1/5 <= ADC Reading < 3/10 full scale | 00\_0000\_0111 |
| 3/10 <= ADC Reading < 2/5 full scale | 00\_0000\_1111 |
| 2/5 <= ADC Reading < 1/2 full scale | 00\_0001\_1111 |
| 1/2 <= ADC Reading < 3/5 full scale | 00\_0011\_1111 |
| 3/5 <= ADC Reading < 7/10 full scale | 00\_0111\_1111 |
| 7/10 <= ADC Reading < 4/5 full scale | 00\_1111\_1111 |
| 4/5 <= ADC Reading < 9/10 full scale | 01\_1111\_1111 |
| 9/10 <= ADC Reading to full scale | 11\_1111\_1111 |

(Note \_ has no significance, it’s for readability only)

**Use of the ATmega4809 RTC timer/counter:**

The RTC timer/counter is used to set and clear a flag variable every 16 seconds.

When the flag variable is ‘0’ and the cylon display is selected the normal cylon pattern (non-inverted) is displayed.

When the flag variable is ‘1’ and the cylon display is selected the inverted cylon pattern is displayed.

**Timer TCA0 setup:**

Use the TCA0.SINGLE specifier because the TCA0 is used in 16-bit mode.

Select normal mode operation by setting TCA0.SINGLE.CTRLB to 0.

You may ignore the TCA0.SINGLE.CTRLC, TCA0.SINGLE.CTRLD, TCA0.SINGLE.CTRLESET/CLR and TCA0.SINGLE.CTRLFSET/CLR registers (leave them at their default settings).

TCA0.SINGLE.CTRLA: Clock source set to divide by 1024, TCA0 enabled.

TCA0.SINGLE.INTCTRL set to Timer Overflow Interrupt enabled.

Set the TCA0.SINGLE.PER register to give the desired timer/counter period.

**ADC0 setup:**

ADC0 AIN3 used

AVDD selected as the ADC0 Reference Voltage

10-bit resolution, Free Running Mode selected, ADC0 enabled

Simple conversion selected (no sample accumulation)

Initial delay set to 16 clocks

ADC0 clock prescaler: 128

ADC Interrupt on Result Ready Enabled

Set the STCONV bit in the ADC0.COMMAND register in the initialisation to begin the first ADC0 conversion.

**RTC Timer setup:**

RTC.CLKSEL set for 1024 Hz internal clock selected.

Use the RTC.PITCTRLA register to select a time-period that will give a 16 second period and also enable the RTC PIT.

Use the RTC.PITINTCTRL to enable the RTC PIT interrupt.

**Hints:**

You may use the code in Example programs as guidelines for your programs.

Use integer arithmetic only. Calculate the ADC reading value that corresponds to 3V and use this value for your comparisons. Set the comparison value using a #define in your code.

Use functions or methods to compartmentalise your code where appropriate.

**Code Structure:**

Initialisation Section:

Initialise relevant variables

Initialise LED array port bits as outputs

Initialise PORTE bit 1 as an input, and enable pull-up resistor on PORTE bit 1

Initialise Timer TCA0

Initialise ADC0

Initialise RTC

Enable global interrupts

while(1)

{

Test PORTE bit 1

if 0 (Button Pressed), Check for new ADC result available

if set, display the ADC0 output in thermometer format

Clear new ADC result available flag

}

Timer TCA0 ISR

{

Test PORTE bit 1

If 1 (Button NOT pressed) Display cylon eyes output on LED array. (Advanced task: check the flag set in the RTC ISR, to decide on setting cylon normal or inverted display).

// Use time setting based on ADC0 result

}

ADC0 ISR

{

Read ADC result and store in a global variable (to be used in main)

Set new ADC result available flag

Set Timer TCA0 ISR time value based on ADC0 result

}

RTC PIT ISR

{

Clear RTC PIT Timer interrupt flag

Change the state of the display mode – cylon or inverted cylon display

}

**Marking Scheme:**

Attendance, initialisation (including variables) and overall commenting style: 2%

Timer TCA0 – LED Array Cylon display code: 2%

ADC0 interrupt, and LED Array bit on/off time setting: 2%

ADC0 output display (thermometer style): 2%

Timed switchover of display from normal cylon to inverted cylon, using RTC Timer: 2%

Total: 10%

**Timetable:**

Demonstrate your program running on Arduino + shield in the appropriate lab session.

Submit the final versions of your programs before Week 8.

# Appendix

# ATmega4809 mapping to Arduino Nano and UNO D0-D13 digital I/O

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arduino UNO | Arduino Nano Every | ATmega4809 Port Pin | Shield Use | ATmega328P |
| D0 | D0 | PORTC bit 4 (PC5) | LED 0 | PORTD bit 0 |
| D1 | D1 | PORTC bit 5 (PC4) | LED 1 | PORTD bit 1 |
| D2 | D2 | PORTA bit 0 (PA0) | LED 2 | PORTD bit 2 |
| D3 | D3 | PORTF bit 5 (PF5) | LED 3 | PORTD bit 3 |
| D4 | D4 | PORTC bit 6 (PC6) | LED 4 | PORTD bit 4 |
| D5 | D5 | PORTB bit 2 (PB2) | LED 5 | PORTD bit 5 |
| D6 | D6 | PORTF bit 4 (PF4) | LED 6 | PORTD bit 6 |
| D7 | D7 | PORTA bit 1 (PA1) | LED 7 | PORTD bit 7 |
| D8 | D8 | PORTE bit 3 (PE3) | Capture | PORTB bit 0 |
| D9 | D9 | PORTB bit 0 (PB0) |  | PORTB bit 1 |
| D10 | D10 | PORTB bit 1 (PB1) |  | PORTB bit 2 |
| D11 | D11 | PORTE bit 0 (PE0) |  | PORTB bit 3 |
| D12 | D12 | PORTE bit 1 (PE1) | Push Btn0 | PORTB bit 4 |
| D13 | D13 | PORTE bit 2 (PE2) | Push Btn1 | PORTB bit 5 |

# ATmega4809 mapping to Arduino Nano and UNO A0-A5 analog Input (and digital I/O)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arduino UNO | Arduino Nano Every | ATmega4809 Port Pin | ATmega4809 Ain | Shield Use |
| A0 | A0 | PD3 | AIN[3] | Potentiometer Out\* |
| A1 | A1 | PD2 | AIN[2] |  |
| A2 | A2 | PD1 | AIN[1] | Potentiometer Out\* |
| A3 | A3 | PD0 | AIN[0] |  |
| A4 | A4 | PA2/SDA |  | LED 8\*\* |
| A5 | A5 | PA3/SCL |  | LED 9\*\* |
|  |  |  |  |  |
| SDA | SDA | PA2/SDA |  |  |
| SCL | SCL | PA3/SCL |  |  |

\* Depending on JP7 setting

\*\*If JP5 and JP6 are connected