# Complex Stepper Motor Driver

## Overview

In this assignment you are required to design and demonstrate a complex stepper motor driver with various inputs and outputs using both your FPGA and MCU hardware kit. The complexity of the design must meet strict specifications that are unique to each student. You are expected to demonstrate a rotating stepper motor with various features and interactive displays, using both the FPGA and MCU hardware kit.

This assignment can be broken down into two parts, the FPGA part and the MCU part. The FPGA part requires you to design a half step excitation stepper motor driver that also displays the rotation direction using an animation on the seven segment display. The direction of rotation is to be easily toggled using a simple click of a switch. The stepper motor wave step excitation is to be sourced from an external clock input that varies with frequency between 100 Hz to 1 kHz.

The MCU part requires you to write a program that has multiple hardware inputs and outputs. All hardware inputs are to come from a standard singular joystick controller that features one digital switch and two analog potentiometers. Three of the hardware outputs are simple LEDs that are available on a "Pixel" Red Green Blue LED module. The fourth and final hardware output is a square wave with a fixed Duty Cycle but variable frequency. The fixed Duty Cycle is unique to each student and the frequency should vary between 100 Hz to 1 kHz, depending on an analog reading of the joystick controller.

Further specifications are defined over the FPGA and MCU specification pages.

## Submission of work

You are first required to submit a softcopy of your report before 18.00 on Wednesday the 17th of October 2018. The report should contain everything you need to explain your design during your demonstration session, which will be that same week. A timetable of demonstration sessions will be posted on the 8th of October.

You are required to demonstrate and verbally explain both the functionality and design process for your solution of the assignment during your scheduled demonstration session. You can think of your report as a demonstration aid. You need to use your demonstration session to show how you completed the work.

## Hardware Requirements

- OMDAZZ FPGA Experimenter Board.
- Altera USB Blaster.
- Arduino UNO atmega328 Experimenter Board.
- USB ASP with 6 Pin Adapter.
- RGB LED Module, Common Cathode.
- Singular Joystick Controller Board.

## Software Requirements

- Quartus Prime Lite version 18 or higher.
- Atmel Studio version 7 or higher.

# Further Overview

## Subject Learning Objectives (SLOs)

As written in the subject outline, this assessment task addresses the following subject learning objectives (SLOs):

1. (1) Design combinational and sequential logic circuits.
2. (2) Implement and evaluate logic circuits using appropriate software tools.
3. (5) Identify basic assembly programming techniques.
4. (6) Construct and implement basic assembly programs.
5. (7) Write and evaluate assembly programs using appropriate software tools.

## Textbook Information

The following assignment has been written after reading the required, recommended and reference textbooks. It is strongly recommended that you first complete the relevant chapters and tutorial questions from the text books before starting this assignment.

- Digital Systems. ISBN: 9781486011964. Compiled by Adel Al-Jumaily and Ahmed Al-Ani, UTS

- The AVR Microcontroller and Embedded Systems Using Assembly and C: Using Arduino Uno and Atmel Studio, Second Edition, Sepehr Naimi, Sarmad Naimi, Muhhamad ali Mazidi Microdigitaled, 2017, ISBN: 9780997925968

# FPGA Specifications

## Stepper Motor Rotation

1. The stepper motor must be driven using half step excitation.

2. The stepper motor must rotate both Clockwise and Anticlockwise without hardware rewiring.

3. The stepper motor excitation must be driven by an external clock that is fed into a singular pin on the FPGA kit. The FPGA kit must have 4 output pins that are wired to the ULN2003 circuit board.

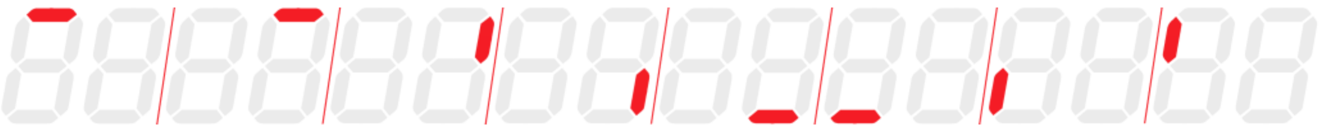
## Rotation Toggling

1. The control for direction of rotation of the stepper motor must be implemented using a toggle button. Clicking and releasing the toggle button will change the rotation direction. The button must not be held down to implement one direction of rotation, the button must be a toggle implementation. Please see the Debouncing Circuit OneNote document [redacted]

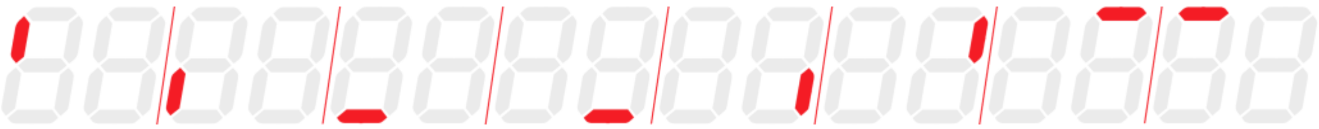
2. The toggle button assignment is unique to each student and defined in the student specification document.

## Seven Segment Display

1. Clockwise rotation of the stepper motor must be indicated using an animation on two of the segments as follows:



2. Anticlockwise rotation of the stepper motor must be indicated using an animation on two of the segments as follows:



3. The rotation animation must be driven by a frequency divided clock source. This clock source must be the same external clock source that is driving the stepper motor excitation. The frequency division is unique to each student and defined in the student specification document. The frequency division must be implemented from scratch using Flip-Flops.

4. While displaying the above rotation animation, the 4th segment must permanently display a specific number. This specific number is unique to each student and defined in the student specification document.

5. The multiplexing implementation to switch between the rotation animation and the permanent number must be sourced from the onboard 50 MHz quartz crystal using a lpm_counter symbol to generate 128 Hz. This is the only Quartus symbol you are allowed to use. All other counters and frequency dividers must be designed from scratch using Flip-Flops. This specific Quartus lpm_counter to output 128 Hz has been covered in a previous tutorial, you can simply copy and paste the implementation.

## 8bit Serial Data Output

1. Using an output pin of the FPGA, you must continuously send out an 8 bit number. This number is unique to each student and defined in the student specification document. The 7th bit is to output first, followed by the 6th, 5th, 4th… 0th.

2. This output must be in sync with the half step stepper motor excitation. There should be 8 stages in this step excitation, use each stage to output a bit of your specific serial number.

# MCU Specifications

## Joystick Input

1. The VRX pin must connect to the atmega328P input pin as defined in the student specification document. This input pin is unique to each student.

2. The VRY pin must connect to the atmega328P input pin as defined in the student specification document. This input pin is unique to each student.

3. The SW pin must connect to the atmega328P input pin as defined in the student specification document. This input pin is unique to each student.

## Square Wave Output

1. You must generate a square wave on the output pin as defined in the student specification document. This output pin is unique to each student.

2. The Duty Cycle of the square wave is fixed and unique to each student. This is defined in the student specification document. Your implementation of the Duty Cycle must be accurate to within 1%.

3. The frequency of the output square wave is to vary with the analog VRY input from the Joystick.
   When VRY = 0 Volts, the output frequency must be 100 Hz. When VRY = 5V, the output frequency must be 1 kHz. There must be a linear correlation of frequency output to the Joystick input. For an input of (0 - 5 V), an output of (100 Hz - 1 kHz) is expected. As an example, when VRY = 2.5V, the output frequency will be 500 Hz.

## LED Indication

1. The input pin SW is to be read by the MCU and when the Joystick is pressed down LED_A must be illuminated. When LED_A is illuminated, LED_B and LED_C are not to be affected. LED_A is unique to each student and defined in the student specification document.

2. The input VRX is to be read by the MCU and when the threshold named Thres_B is reached, LED_B must be illuminated. When Thres_B is not reached, LED_B is not illuminated. When LED_B is illuminated or de-illuminated, LED_A and LED_C are not to be affected. Thres_B and LED_B are unique to each student and defined in the student specification document.

3. The input VRX is to be read by the MCU and when the threshold named Thres_C is reached, LED_C must be illuminated. When Thres_C is not reached, LED_C is not illuminated. When LED_C is illuminated or de-illuminated, LED_A and LED_B are not to be affected. Thres_C and LED_C are unique to each student and defined in the student specification document.

# Student Specifications

[redacted]

## Toggle Button
This is the physical button on your FPGA board that will toggle the stepper motor direction of rotation.

## Permanent Number
This is the number that must be displayed permanently on the 4th digit of the FPGA board seven segment display.

## Frequency Divider
This is the number that you are required to divide the square wave frequency input by on the FPGA board.

## Serial Number
This is the 8bit number that you are to constantly send out on any pin of your choice on the FPGA board.

## JoyVRX
This is the analog input pin on your atmega328P board that you are required to wire your joystick to.

## JoyVRY
This is the analog input pin on your atmega328P board that you are required to wire your joystick to.

## JoySW
This is the digital input pin on your atmega328P board that you are required to wire your joystick to.

## Frequency Out Pin
This is the output pin on your atmega328P board that you are required to implement the square wave on.

## Duty Cycle
This is the fixed duty cycle of your output square wave on the atmega328P board.

## LED_A
This the colour of the LED that you are to illuminate.

## LED_B
This the colour of the LED that you are to illuminate.

## LED_C
This the colour of the LED that you are to illuminate.

## Thres_B
This is your threshold condition to illuminate LED_B

## Thres_C
This is your threshold condition to illuminate LED_C

# Assignment Guidelines

## Report Cover Sheet

[redacted]

## Completing the Assignment

- Individual implementation of the assignment is required.
- Any student found copying material or plagiarising will be awarded a mark of zero and may face disciplinary action.
- PROTECT YOUR WORK: Do not give other students any part of your assignment.
  Students found sharing work will be awarded a mark of zero and may face disciplinary action.
  If your friends are encountering difficulty with part of the assignment, help them by explaining the theory behind the problem, or refer them to a source where they can get further information, but DON'T give them your report, simulations or source code.

## Due Date

- This assignment is due on Wednesday the 17th of October at 18:00.
- You are required to submit a softcopy report to Turnitin ([redacted] before 18:00 on Wednesday the 17th of October. This softcopy report will be opened by a tutor during your demonstration session and you will be expected to use this report to facilitate your demonstration.
- For Thursday the 18th of October and Monday the 22nd of October, your lab activity session will be replaced with a scheduled demonstration time. You are expected to arrive at this scheduled time and demonstrate your solution to this assignment using both your Turnitin report and hardware kit.
- **No alternative demonstration times will be scheduled. No extension will be given for the softcopy report submission to Turnitin.** It is your responsibility to ensure you have submitted your soft copy report to Turnitin before the deadline. It is strongly recommended that you submit the report 24 hours before it is due.

## FPGA Implementation and Demonstration

- The entire digital logic design must be implemented only with schematics.
- The full schematics of your design must be presented in your Turnitin report.
- You must demonstrate the proper design processes for sequential and combinational logic circuits for each part of the digital logic design, even if you are using the provided circuits or circuits from the textbook. Failure to show adequate design detail in the report will result in 0 marks being awarded, even if your demonstration is working.
- You are **not permitted to** use inbuilt counter symbols in Quartus. Counters, frequency dividers and other circuits must be designed from scratch. The only exception to this is the 50 MHz quartz crystal to 128 Hz lpm_counter that is discussed on the specification page.
- You are **allowed** to use built-in flip-flops and logic gates.
- You will be expected to verbally explain the truth tables and circuit schematics in your Turnitin report during the demonstration of your assignment.

## MCU Implementation

- All code on the atmega328P must be implemented in assembly code. You are not permitted to use any other higher level programming language.
- The full source code listing of your firmware **must be presented** in your Turnitin report. You will be expected to verbally explain the firmware in your Turnitin report during the demonstration of your assignment.

## Hardware Implementation Marking Scheme

### FPGA

| | |
|---|---|
| Stepper Motor Rotation | 3 Marks |
| Rotation Toggling | 1 Mark |
| Seven Segment Display | 4 Marks |
| 8bit Serial Data Output | 2 Marks |

### MCU

| | |
|---|---|
| Joystick Input | 1 Mark |
| Square Wave Output | 5 Marks |
| LED Indication | 4 Marks |

## Evaluation Criteria

The following table of criteria will be used by tutors for evaluating responses to questions and components of your demonstration:

| **Evaluation** | **Mark (%)** | **Description** |
|---|---|---|
| Excellent | 100 | All relevant material is presented in a logical manner showing clear understanding, and sound reasoning. |
| Good | 75 | Nearly all relevant material is presented with good organisation and understanding. |
| Acceptable | 50 | Most relevant material is presented with acceptable organisation and understanding. |
| Poor | 25 | Little relevant material is presented and/or poor organisation or understanding. |
| No attempt | 0 | No attempt. |

## Students with difficulty meeting assessment requirements

[redacted]

Note also that students may apply for special consideration because of illness or other circumstances (not work related) beyond their control. The “Application for Special Consideration form” has a section that must be filled in by a doctor, counsellor or other relevant professional authority. A medical certificate alone is not adequate and will not be accepted.

Note that it is up to the students to provide adequate information about their circumstances. University staff will not chase additional information and the Subject Coordinator has the right to reject applications that lack sufficient information.

It is the student’s responsibility to contact the Subject Coordinator to find out what action has been taken and to obtain details of any additional assessment required or learning and assessment special arrangements.

[redacted]