CURRENT CONTROL IN INDUCTOR / SOLENOID USING TMS320F2837xD

A project report submitted for the fulfilment of academic requirements for the award of Degree in Bachelor of Engineering in the Department of Electrical and Electronics Engineering

Submitted by:

THOMAS GEORGE 4NI20EE095

Under the Guidance of,

Dr. S Gurumurthy
Assistant Professor
Department of Electrical and Electronics Engineering
The National Institute of Engineering

Department of Electrical & Electronics Engineering
The National Institute of Engineering
(Autonomous Institute under VTU, Belagavi)
Mysuru – 570 008

2022-23

The National Institute of Engineering

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Manandavadi Road, Mysuru – 570 008, Karnataka
Recognized by AICTE, New Delhi
Accredited by National Board of Accreditation, New Delhi Grant-in-Aid by the Government of Karnataka

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

CERTIFICATE

Certified that this minor project report entitled **"CURRENT CONTROL IN INDUCTOR / SOLENOID USING TMS320F2837xD"** *is a bonafide work carried out by* ***THOMAS GEORGE*** *(4NI20EE095),under the guidance of* ***Dr. S Gurumurthy****, Professor in partial fulfilment for the award of the degree of Bachelor of Engineering in Electrical and Electronics Engineering at The National Institute of Engineering (Autonomous Institute under VTU) during the academic year 2022-2023.*

Dr. S Gurumurthy	**Dr. H Pradeepa,** HoD,
Professor	
Department of E&E	Department of E&E
The National Institute of Engineering, Mysuru	The National Institute of Engineering, Mysuru

DECLARATION

I, THOMAS GEORGE bearing USN- **4NI20EE095** student of **6th** semester of UG program, from the Department of **Electrical and Electronics Engineering**, hereby declare that the project work entitled "**CURRENT CONTROL IN INDUCTOR / SOLENOID USING TMS320F2837xD"** has been carried out by me under the guidance of **Dr.S Gurumurthy, Prof. EEE Department**. This project work is submitted to **The National Institute of Engineering**, Mysuru, (An Autonomous institute under VTU, Belagavi) in partial fulfillment of the course requirements for the award of a degree in **Electrical and Electronics Engineering** during the academic year 2022-2023. This written submission represents a record of original work and I have adequately cited and referenced the sources.

PROJECT TEAM

NAME OF THE STUDENT	USN	Signature
1. THOMAS GEORGE	4NI20EE095	

ACKNOWLEDGEMENT

I convey my gratitude to all who have made valuable contribution to the Minor project throughout. I wish to express my profound thanks to all those who have encouraged me in doing this project.

I would like to thank our guide Dr. S Gurumurthy, Professor providing for their continuous guidance and support in developing the Minor project. I thank him for his thoughtful suggestions and clear advice.

I also express our heartfelt gratitude to our Head of the department, Dr.Pradeepa H for providing me with all the facilities and permission to do this Minor project.

I consider myself privileged to have studied in esteemed institution, THE NATIONAL INSTITUTE OF ENGINEERING, Mysuru. Lastly, I would like to thank all the faculty and staff members of the Department of Electrical and Electronics Engineering, Mysuru.

ABSTRACT

The converter architecture consists of a high-frequency switching circuit, a control module, and a low-pass filter. The switching circuit employs a high-frequency power semiconductor device, such as a MOSFET or an IGBT, to rapidly switch the input voltage across the inductor. This switching action enables the creation of a pulsed DC current waveform in the inductor, which is essential for many inductive load applications.

The control module plays a vital role in regulating the converter operation. It includes a feedback loop that continuously monitors the inductor current and adjusts the switching frequency and duty cycle accordingly. This feedback mechanism ensures precise control over the output current, preventing excessive current ripple and ensuring stable operation of the inductive load.

To mitigate the high-frequency switching noise and harmonics generated by the converter, a low-pass filter is incorporated. The filter attenuates the undesired highfrequency components, providing a clean DC current to the inductor. This helps minimize electromagnetic interference and enhances the overall performance of the system.

The proposed converter exhibits several advantages over traditional methods. It offers improved efficiency, reduced power losses, and enhanced controllability, enabling optimal power delivery to the inductive load. Moreover, it provides robust protection features, including overcurrent and overvoltage protection, ensuring safe operation of the system.

Simulation and experimental results demonstrate the effectiveness of the converter in driving a DC current inductor. The system achieves high power conversion efficiency, excellent load regulation, and low output ripple. These results validate the converter's suitability for a wide range of applications requiring precise and reliable control of DC current inductors.

TABLE OF CONTENTS

Details **Page No.**

TABLE OF FIGURES

Details **Page No.**

LIST OF TABLES

Details **Page No.**

LIST OF ACRONYMS

AC Alternating Current

DC Direct current
IC Integrated Circuit
MOSFET Metal Oxide Semiconductor Field Effect Transistor
OP-AMP Operational Amplifier
PWM Pulse width Modulation
IDE Integrated Development Environment

INTRODUCTION

Current Control Concept Using Hysteresis Loop:

Hysteresis current control is a technique commonly used for current regulation in solenoid applications. It is based on the hysteresis loop principle, which provides a simple and effective method to maintain the current within a desired range.

1. Hysteresis Loop:
A hysteresis loop represents the relationship between the input signal and the output response in a control system. In the case of current control, the hysteresis loop is used to define the desired current range and the corresponding switching actions.

2. Hysteresis Comparator:
The core component of hysteresis current control is the hysteresis comparator. The comparator compares the measured current value with upper and lower thresholds to determine the control action. It generates a control signal that switches the MOSFET based on the current position within the hysteresis loop.

3. Upper and Lower Thresholds:
The upper and lower thresholds define the desired current range within the hysteresis loop. These thresholds are set based on the application requirements and the characteristics of the solenoid. When the measured current exceeds the upper threshold, the comparator triggers a control action to reduce the current. Conversely, when the measured current falls below the lower threshold, the comparator triggers a control action to increase the current.

4. Hysteresis Band:
The hysteresis band is the region between the upper and lower thresholds in the hysteresis loop. It represents the allowable range of current variation. The width of the hysteresis band determines the level of current stability. A wider hysteresis band allows for more current variation, while a narrower band provides tighter current regulation.

5. Control Action:
When the measured current exceeds the upper threshold, the hysteresis comparator generates a control signal to switch off the MOSFET. This action interrupts the current flow through the solenoid, reducing the current. On the other hand, when the measured current falls below the lower threshold, the comparator generates a control signal to switch on the MOSFET, allowing current to flow through the solenoid and increasing the current.

6. Advantages of Hysteresis Current Control:

- Simple and intuitive concept: The hysteresis loop and comparator approach provide a straightforward method for current regulation.
- Fast response: The hysteresis comparator triggers control actions immediately when the current crosses the thresholds, leading to quick adjustments and accurate current regulation.
- Robustness: Hysteresis current control is less sensitive to noise and disturbances compared to other control techniques, as it operates based on the instantaneous current value rather than feedback or error calculations.
- Self-adjusting: The hysteresis loop automatically adjusts the control actions based on the current position within the loop, providing self-regulating behavior.

By implementing hysteresis current control in the TMS320F2837xD microcontrollerbased system, we can achieve precise current regulation for solenoids. The hysteresis loop concept, combined with the comparator and appropriate threshold settings, allows for efficient control of the solenoid current, ensuring optimal performance and stability.

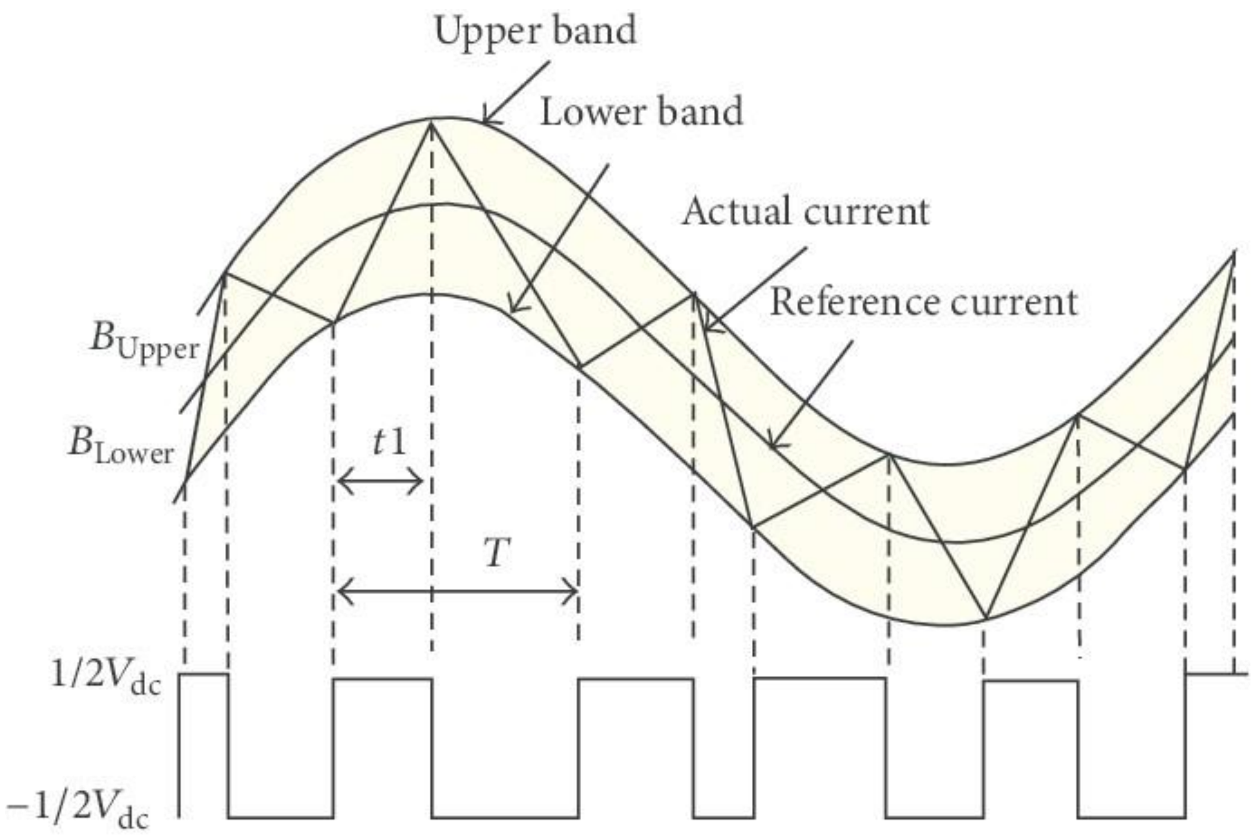


Figure 1- Concept of Current control

TMS320F2837XD - DUAL CORE MICROCONTROLLER

The TMS320F2837xD is a dual-core microcontroller from Texas Instruments that is specifically designed for real-time control applications. It combines the power of a highperformance digital signal processor (DSP) core with a separate control law accelerator (CLA) core. This dual-core architecture enables efficient and parallel execution of control algorithms, making it suitable for demanding control applications.

Key features and characteristics of the TMS320F2837xD microcontroller include:

1. Dual-Core Architecture:

The TMS320F2837xD microcontroller features two independent cores:

- C28x DSP Core: The C28x core is a 32-bit floating-point DSP core that provides highperformance processing capabilities for real-time control tasks.
- CLA (Control Law Accelerator) Core: The CLA core is an additional 32-bit fixed-point processing unit that runs parallel to the C28x core. It is dedicated to executing control algorithms independently, offloading the main DSP core and increasing overall system performance.

2. High-Speed Processing:

The TMS320F2837xD microcontroller operates at high clock frequencies, allowing for fast and efficient execution of control algorithms. The C28x core can reach frequencies up to 200 MHz, while the CLA core can operate at frequencies up to 200 MHz as well.

3. Integrated Peripherals:

The microcontroller includes a wide range of integrated peripherals, including:

- Analog-to-Digital Converters (ADCs): High-resolution ADCs for accurate analog signal acquisition.
- Pulse Width Modulation (PWM) Modules: Dedicated PWM modules for generating precise digital control signals for motor control and power electronics applications. - Communication Interfaces: UART, SPI, I2C, CAN, and Ethernet interfaces for communication with other devices.
- Timers and Interrupts: Timers for precise timing and event management, and interrupt controllers for handling interrupts from various sources.

4. Memory:

The TMS320F2837xD microcontroller provides on-chip memory resources, including:

- Program Memory: Flash memory for storing program code.
- Data Memory: RAM for data storage and variable manipulation.

- C28x Local Data RAM: Dedicated RAM for the C28x core. - CLA Data RAM: Dedicated RAM for the CLA core.

5. Real-Time Control Peripherals:
The microcontroller offers specialized peripherals for real-time control applications, such as high-resolution PWM modules, eQEP (Enhanced Quadrature Encoder Pulse) interfaces for motor control, and dedicated ePWM (Enhanced Pulse Width Modulation) modules for power conversion and motor control.

6. Development Tools and Software:
Texas Instruments provides a comprehensive software development ecosystem for the TMS320F2837xD microcontroller. This includes integrated development environments (IDEs), compilers, debuggers, and libraries, enabling efficient software development and debugging.

The TMS320F2837xD microcontroller's dual-core architecture, high-speed processing capabilities, integrated peripherals, and robust development tools make it a powerful choice for real-time control applications, including motor control, power electronics, renewable energy systems, robotics, and industrial automation. Its versatility and performance make it suitable for both simple and complex control algorithms, enabling precise and efficient control in various applications.

Figure 2 - TMS320F2837xD

OBJECTIVES

The objective of the project on current control in a solenoid using the TMS320F2837xD dualcore microcontroller can be defined as follows:

1. Design and Implement a Current Control System: The main objective is to design and implement a control system that regulates the current flowing through a solenoid. The TMS320F2837xD microcontroller will be utilized to monitor the current and generate control signals to adjust the switching of the MOSFET, ensuring the desired current level is maintained.

2. Achieve Precise Current Regulation: The project aims to achieve precise and accurate current regulation in the solenoid. By utilizing the capabilities of the microcontroller and appropriate control algorithms, the system should be capable of tightly controlling the current within a specific range, minimizing variations and ensuring consistent performance.

3. Optimize Power Efficiency: Another objective is to optimize the power efficiency of the solenoid system. By implementing effective current control techniques, the project aims to minimize power losses and improve overall system efficiency. This may involve reducing the power dissipation in the MOSFETs, optimizing the control algorithm, and minimizing current ripple.

4. Ensure Reliable and Safe Operation: The project aims to ensure the reliable and safe operation of the solenoid system. The current control system should be designed to handle varying load conditions, external disturbances, and potential fault scenarios. Safety features, such as overcurrent protection, should be incorporated to protect the solenoid and associated components.

5. Demonstrate the Capabilities of the TMS320F2837xD Microcontroller: The project seeks to showcase the capabilities of the TMS320F2837xD microcontroller in real-time control applications. By successfully implementing the current control system using the microcontroller's dual-core architecture, integrated peripherals, and development tools, the project aims to highlight its performance, efficiency, and flexibility.

6. Documentation and Presentation: A key objective is to document the project thoroughly and present the findings and outcomes in a comprehensive manner. This includes preparing a project report that covers the theoretical background, methodology, implementation details, experimental results, and conclusions. The report should serve as a valuable resource for

understanding the current control system and its application using the TMS320F2837xD microcontroller.

By achieving these objectives, the project aims to contribute to the field of current control in solenoids and demonstrate the effectiveness of using the TMS320F2837xD microcontroller in real-time control applications.

BLOCK DIAGRAM

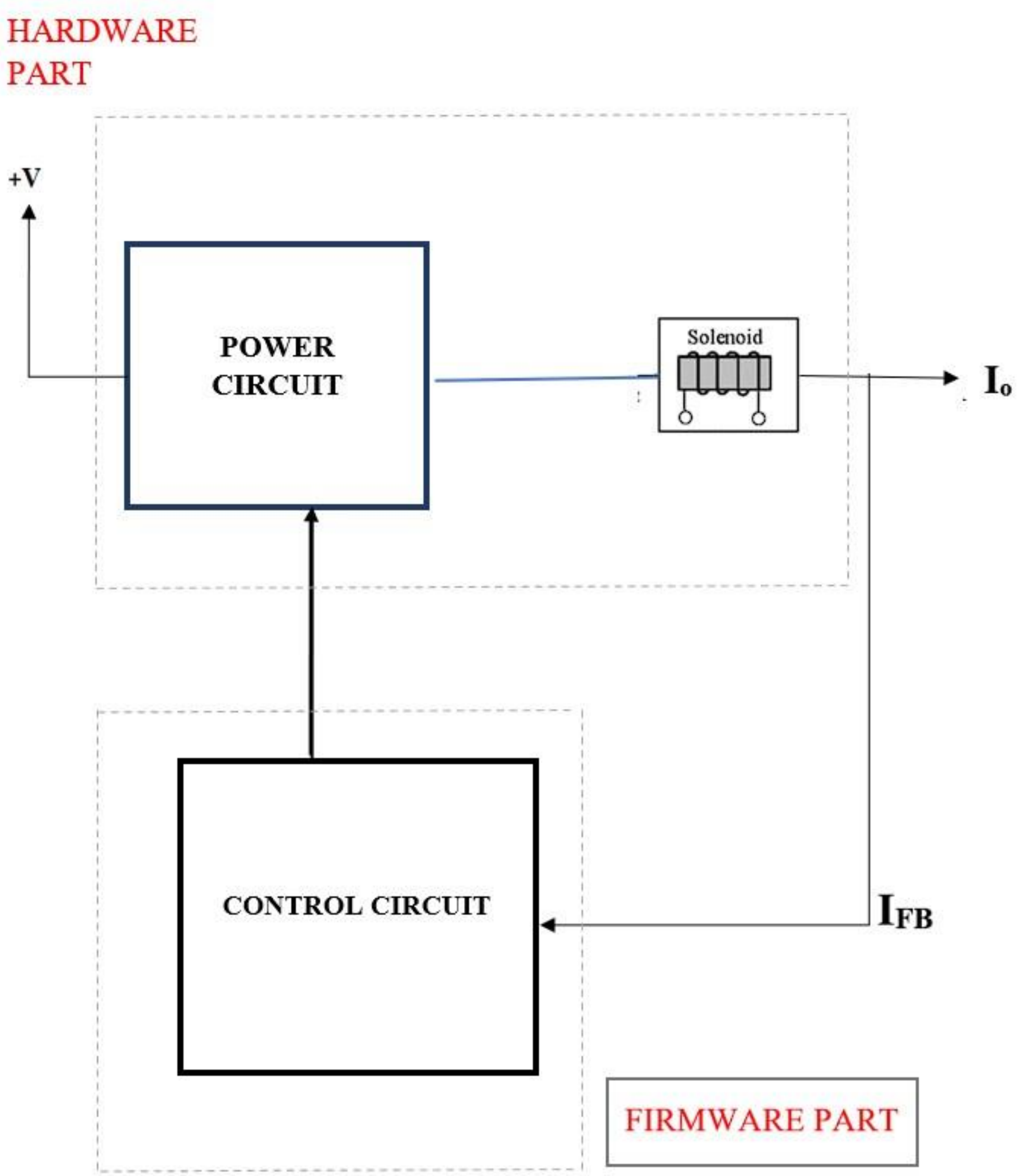


Figure 3 - Block Diagram

The block diagram for the current control system using the TMS320F2837xD microcontroller can be divided into two main parts: the hardware part and the firmware part.

1. Hardware Part:
The hardware part of the system consists of various components that are responsible for the physical implementation of the current control system. These components include:
Solenoid: The solenoid is the load device in the system and represents the actuator that requires precise current control. It could be a linear solenoid, proportional solenoid, or any other type of solenoid device.

Current Sensing Circuit: This circuit is responsible for measuring the current flowing through the solenoid. It typically includes a current sensor, such as a shunt resistor or a current transformer, and conditioning circuitry to convert the current signal into a voltage signal.

MOSFET Switching Circuit: The MOSFET switching circuit controls the current flow through the solenoid. It typically consists of two MOSFETs: one for switching the current on and off and the other as a freewheeling diode across the solenoid. The switching circuit is controlled by the microcontroller's GPIO pins.

Power Supply: The power supply provides the necessary voltage and current to the solenoid and other components in the system. It should be designed to meet the power requirements of the solenoid and provide stable and regulated power.

2. Firmware Part: the firmware part includes the System PLL Clock, ADC, CPU Timer, Interrupts, and GPIO. Here's an updated description:

1. System PLL Clock:
The firmware initializes and configures the System PLL Clock of the TMS320F2837xD microcontroller. This involves selecting the clock source, setting the PLL multiplication and division factors, and ensuring the PLL is locked and stable. The System PLL Clock provides the main clock source for the microcontroller and ensures accurate timing for the firmware operations.

2. ADC (Analog-to-Digital Conversion):
The firmware configures and utilizes the built-in ADC module of the microcontroller. It sets the ADC clock frequency, resolution, and signal mode. The ADC is responsible for converting the analog current measurement from the current sensing circuit into a digital

value. The firmware configures the ADC channels, triggers conversions, and reads the converted digital values for further processing.

3. CPU Timer:
The firmware utilizes the CPU Timer module of the microcontroller for various timingrelated tasks. It configures the CPU Timer, sets the prescaler, and defines the period register value. The CPU Timer can be used for generating precise timing intervals for executing the control algorithm, triggering ADC conversions, or performing other timedependent operations in the firmware.

4. Interrupts:
The firmware configures and handles interrupts to respond to specific events and ensure timely execution of critical tasks. It enables and sets up interrupts for the ADC module, allowing the firmware to be notified when ADC conversions are completed. Additionally, the firmware may configure other interrupts as needed, such as timer interrupts for periodic control algorithm execution or GPIO interrupts for external events.

5. GPIO (General Purpose Input/Output):
The firmware controls the MOSFET switching circuit and interacts with other external devices using the GPIO pins of the microcontroller. It initializes and configures the GPIO pins as either input or output, sets pin directions, and manipulates GPIO registers to control the MOSFETs. By manipulating the GPIO pins, the firmware can turn the MOSFETs on or off based on the control algorithm's output and the desired current level.

These firmware components work together to implement the current control system using the TMS320F2837xD microcontroller. The System PLL Clock provides the necessary timing and synchronization, the ADC converts analog current measurements to digital values, the CPU Timer ensures accurate timing for control operations, interrupts enable timely response to events, and GPIO allows control of the MOSFET switching circuit and communication with external devices.

CIRCUIT DIAGRAM

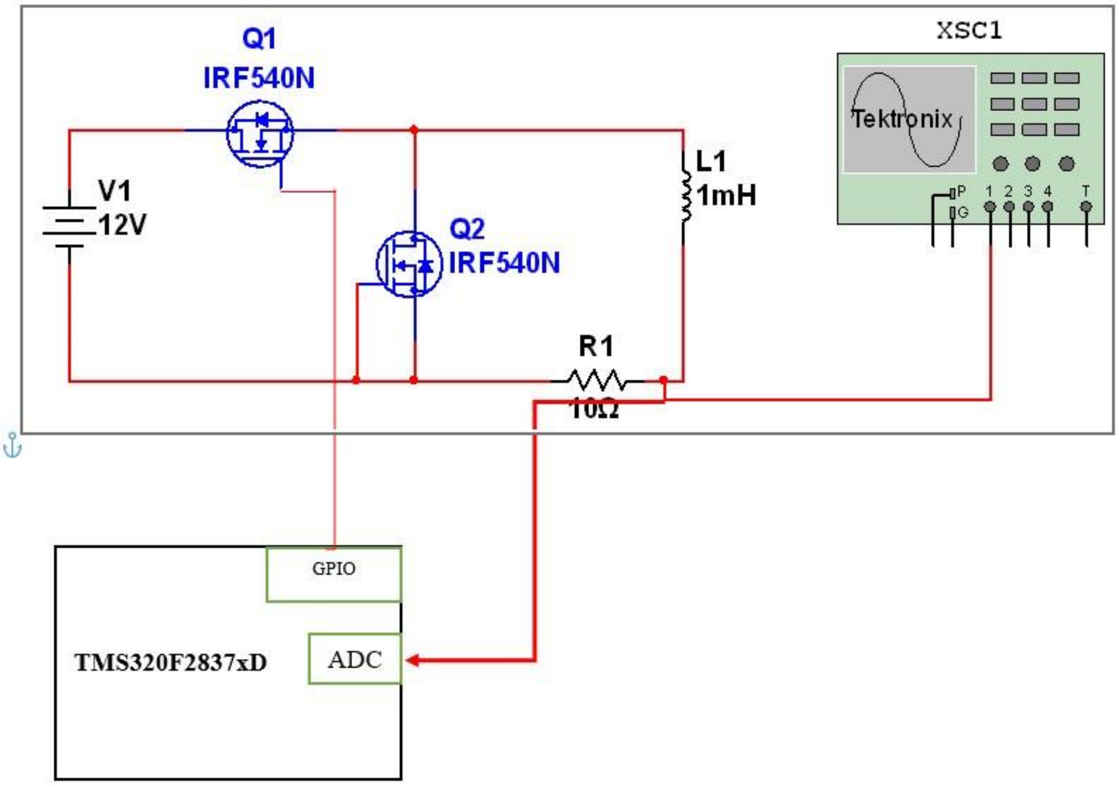


Figure 4 - Circuit Diagram

WORKING PRINCIPLE

The working principle of the solenoid current control circuit with the given control algorithm is

1. Power Supply: The power supply provides the necessary voltage and current to the solenoid and other components in the circuit.
2. Current Sensing: The current sensing circuit, which includes a current sensor and conditioning circuitry, measures the current flowing through the solenoid. The current sensor converts the current into a voltage signal that can be accurately measured.
3. ADC Conversion: The voltage signal from the current sensing circuit is connected to the ADCA2_IN ADC input pin of the microcontroller. The microcontroller's firmware configures and initiates the ADC conversion to convert the analog voltage signal into a digital value.
4. Control Algorithm: The firmware running on the microcontroller implements a control algorithm to regulate the current flowing through the solenoid. In this case, the control algorithm compares the measured current value (obtained from the ADC conversion) with two threshold values: 1700 and 2500.

5. GPIO Control:

- If the measured current value (j) is less than 1700, the firmware sets the GPIO16 pin to logic high using the statement `GpioDataRegs.GPASET.bit.GPIO16 = 1;`. This turns on the MOSFET switch, allowing current to flow through the solenoid.

- If the measured current value (j) is greater than 2500, the firmware clears the GPIO16 pin to logic low using the statement `GpioDataRegs.GPACLEAR.bit.GPIO16 = 1;`. This turns off the MOSFET switch, interrupting the current flow through the solenoid.

6. Freewheeling Diode: The MOSFET switching circuit includes a freewheeling diode connected across the solenoid. This diode provides a path for the current to flow when the MOSFET switch is turned off, preventing voltage spikes and protecting the circuit components.

7. Control Loop: The entire process of measuring the current, comparing it with the threshold values, and controlling the MOSFET switching occurs in a closed control loop. The firmware continuously samples and updates the control signals based on the feedback from the ADC conversion.

By adjusting the MOSFET switching based on the control algorithm's output, the solenoid current control circuit ensures that the solenoid current remains within the desired range. When the current is below 1700, the MOSFET switch is turned on, allowing current to flow through the solenoid. When the current exceeds 2500, the MOSFET switch is turned off, interrupting the current flow.

the threshold values (1700 and 2500) and the GPIO pin (GPIO16) mentioned here are specific to the provided control algorithm. You may adjust these values or GPIO pins based on your specific requirements and circuit configuration.

- When the switch (the transistor) turns on, current I_L flows in the inductor L, and energy is stored.
- At this time, the diode is turned off.
- The inductor current I_L is expressed by the following equation.

Where ton is ON time

$$I_L = \frac{Vin - Vout}{L} \times ton$$

- When the switch (the transistor) turns off, the energy stored in the inductor is output through the diode.
- At this time, the switch (the transistor) is OFF.
- The inductor current It is expressed by the following equation.

Where toff is OFF time

$$I_L = \frac{Vout}{L} \times toff$$

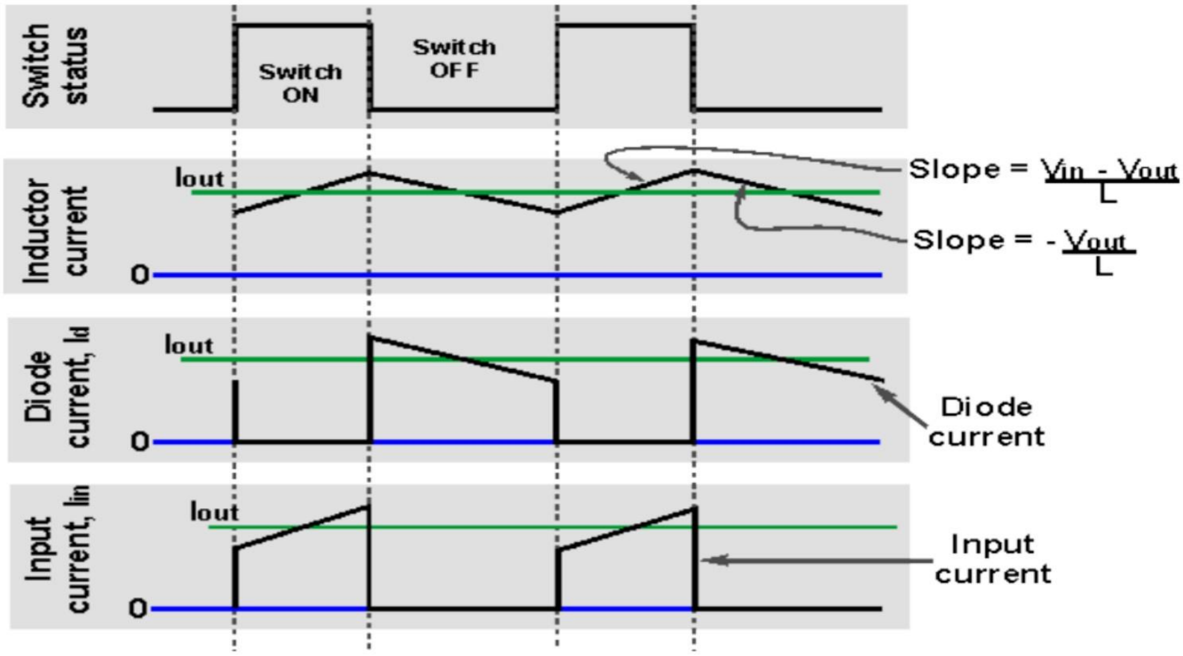


Figure 5 - Graph of Variation in Current in the Inductor

Discontinuous Mode and Continuous Mode

In switching operation, there are two modes, a discontinuous mode and a continuous mode. They are compared in the following table. The "operation" item for comparison is the waveform of the currents flowing in the primary windings and secondary windings of the transformer. In discontinuous mode, there is a period in which the inductor current IL is interrupted, hence the name, discontinuous mode. In contrast, in continuous mode there is no period in which the inductor current is zero.

In the case of the continuous mode, when the switches are ON, a reverse current flows during the reverse recovery time (trr) of the rectifying diode, and losses occur due to this reverse current. In low voltage switching DC/DC conversion, the reverse voltage of the rectifying diode is low and the reverse current is also small, and so generally the continuous mode is used, giving priority to reducing the output ripple voltage and harmonics. However, in AC/DC conversion, the diode reverse voltage is high and a large reverse current flows, and so discontinuous mode, in which a reverse current does not flow and losses are reduced, is generally used. However, the peak current becomes large, and when the load is large, sometimes operation in continuous mode is preferred.

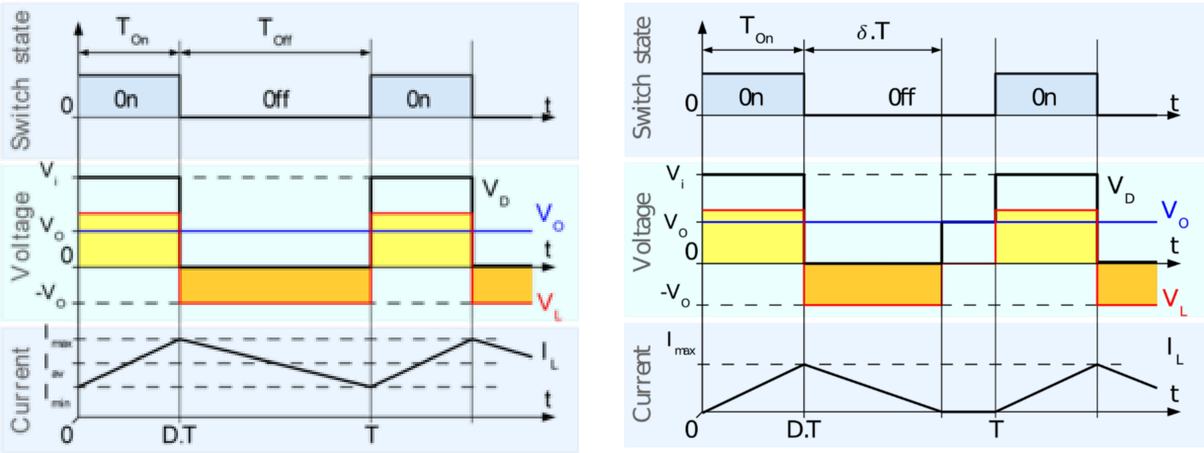
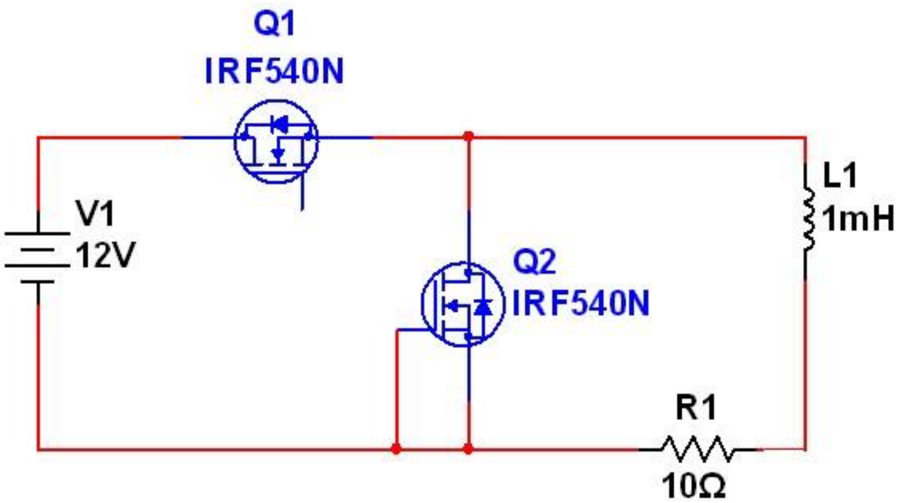


Figure 6 - Graph of Continuous Mode and Discontinuous Mode

<u>COMPONENTS</u>

COMPONENTS	SPECIFICATIONS	QUANTITY
MOSFET	**IRF540N**	2
Inductor	**1mH**	1
Resistor	**10Ω**	1
TMS320F2837xD	**-**	1

Oscilloscope	**Tektronix**	1
Code Composer Studio	**IDE**	-



POWER CIRCUIT

Figure 7 - Power Circuit

The power circuit plays a crucial role in providing the necessary power to drive the solenoid and other associated components. The power circuit includes a MOSFET for switching and another MOSFET acting as a freewheeling diode across the inductor, as well as a small load resistance. Here's an explanation of the power circuit components and their functions:

1. Power Supply:
- DC Power Source: The power circuit requires a DC power source to provide the required voltage for the system. This can be a DC power supply or a battery.

2. MOSFET for Switching:
- Switching MOSFET: The MOSFET serves as a switch in the power circuit. It controls the current flow through the solenoid by turning on and off based on the control signals from the microcontroller.
- Gate Control: The gate of the switching MOSFET is connected to a GPIO pin of the TMS320F2837xD microcontroller. The microcontroller provides control signals to the MOSFET gate to control its switching behaviour.

3. Freewheeling Diode:
- MOSFET as Freewheeling Diode: The second MOSFET is used as a freewheeling diode across the inductor. When the switching MOSFET turns off, the current flowing through the inductor needs a path to circulate and avoid voltage spikes. The freewheeling MOSFET allows the current to flow through it, providing a path for the inductor current.

4. Inductor and Load Resistance:
- Inductor: The inductor is a key component in the power circuit. It stores energy in its magnetic field when the switching MOSFET is turned on and releases this energy when the MOSFET is turned off. The inductor helps regulate the current flowing through the solenoid.
- Load Resistance: The small load resistance represents the load connected to the solenoid. It can be a resistor or the solenoid coil itself. The load resistance determines the current flowing through the solenoid and affects its operation.

5. Protection Mechanisms:
- Current Limiting: To protect the solenoid and the circuit, current-limiting components like resistors or current-sensing devices can be included in series with the solenoid. They limit the maximum current that can flow through the solenoid, preventing damage due to excessive current.

Overall, the power circuit in your project controls the switching of the solenoid through the MOSFET, provides a path for the inductor current using the freewheeling MOSFET, and includes a load resistance to determine the current flowing through the solenoid. Proper design and configuration of the power circuit are essential to ensure efficient and controlled operation of the solenoid and protect the components from excessive current or voltage spikes.

HARDWARE DESIGN:

INDUCTOR DESIGN:

Voltage ripple = 2%, Current ripple = 30%

P = 100W, V_{in} = 36v, V_{out} = 24v to 48v

$D_{min} = \frac{24+0.7}{36+24+0.7} = 0.4069$

$D_{max} = \frac{48+0.7}{36+48+0.7} = 0.5744$

$\Delta I_L = \frac{Iout \times Vout}{Vin(min)} \times 30\% \quad \frac{100}{36} = \times 0.3 = 0.8333$

$L_1 = L_2 = \frac{Vin(min)}{\Delta I_L} \times Dmax \quad 0. \quad \frac{36}{8333 \times 25000} = \times 0.5749 = 0.9934mH$

For the current controller ckt design, inductor of inductance L= 1mH carrying a peak current of 2.77A is required. The first step of inductor design is selection of ferrite core.
For the selection core, the product of area of the core AC and area of the window AW is needed. The product is calculated using

$$A_C A_W = \frac{L\, Ip\, Irms}{Bm\, Kw\, J}$$

Where:

I_p - Peak Current

I_{rms} - RMS value of the peak current

B_m - Maximum Flux Density, For ferrite core Bm= 0.2×10^{-6} wb/mm^2

K_w - Winding Factor Kw=0.5

J - Current Density, for copper wire J = 8 A/mm^2

$$Ap = AcAw = \frac{L \times Ip \times Irms}{Bm \times Kw \times J} = \frac{993.4 \times 10^{-6} \times 3.22 \times 2.91}{0.25 \times 0.4 \times 3 \times 10^{6}} = 31{,}027.85$$

From the core selector chart of FDK ferrite core, we select EE4215 with the specification

Ac = 181mm^2

The Number of Turns of the Inductor coil,

$$N = \frac{L \times Ip}{Bm \times Ac} = \frac{993.4 \times 10^{-6} \times 3.22}{0.25 \times 181 \times 10^{-6}}$$

= 71

N = 72 for EE4215, Here N is number of Turns

Approximately, 72 turns are wound around the core. Based on the equation aw= Irms/J, SWG 18 copper wire is chosen as the windings. From the calculations air gap obtained is 0.23mm.

Length of air gap, Ig = $\frac{\mu 0 \times N \times Ip}{Bm}$

CIRCUIT IMPLEMENTATION PART

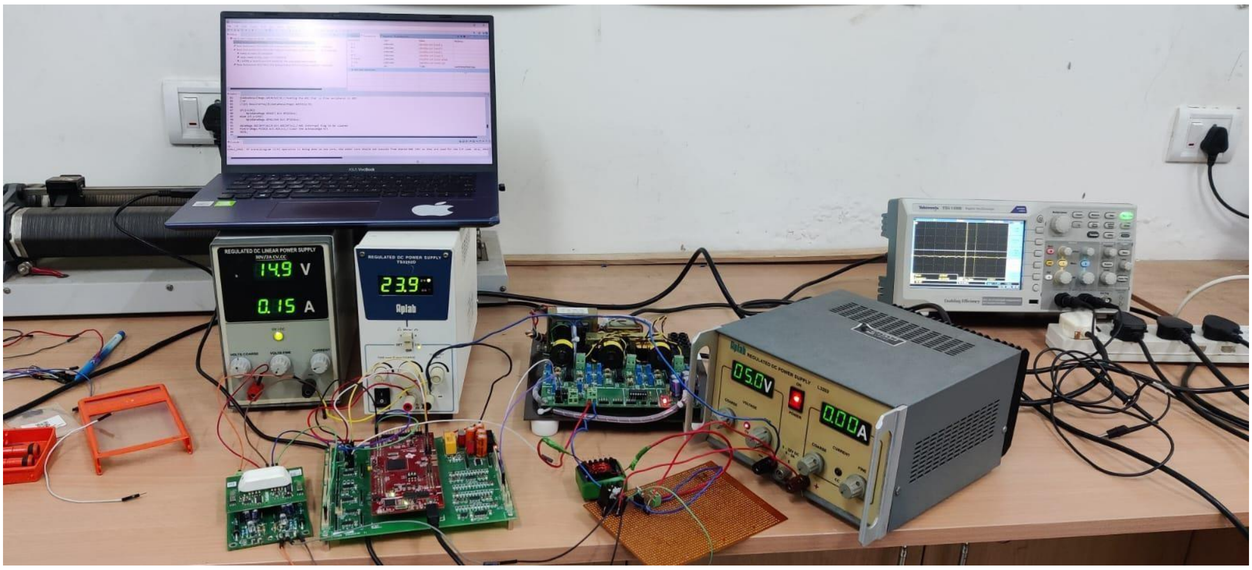


Figure 8 - Circuit Implementation Part

RESOURCE PLANNING

Resource planning for the TMS320F2837xD microcontroller involves effectively managing and utilizing the available resources such as the ADC input, GPIO, CPU Timer, and the ADCISR (ADC Interrupt Service Routine). Resource planning for each of these components:

1. ADCA2_IN ADC:
- Resource Allocation: Allocate the ADCA2_IN ADC input pin for connecting the current sensing circuit to measure the solenoid current. Configure the ADC module to use the ADCA2_IN input channel for analog-to-digital conversion.
- Sampling Rate: Determine the required sampling rate based on the system requirements and the dynamics of the solenoid current. Ensure that the ADC sampling rate is sufficient to capture the desired level of accuracy and responsiveness.
- Resolution and Precision: Configure the ADC resolution and precision based on the desired level of accuracy. The TMS320F2837xD microcontroller offers different ADC resolution options to balance between accuracy and processing overhead.
- ADC Configuration: Set up the ADC control registers, including reference voltage selection, sampling window size, conversion triggers, and acquisition time. Ensure that the ADC is properly configured for accurate and reliable conversion of the analog current signal.

2. GPIO16:
- Pin Configuration: Configure GPIO16 as an output pin to control the MOSFET switch in the solenoid circuit. Set the pin direction, mode, and other settings as per the system requirements.
- Logic Levels: Determine the logic levels for GPIO16 that correspond to the desired MOSFET switch state (on or off). Ensure that the GPIO16 pin is properly set to the intended logic level to control the MOSFET switch effectively.

3. CPU Timer0:
- Resource Allocation: Allocate and configure Timer0 for specific timing requirements in the system. Timer0 can be used for various purposes, such as generating interrupts at specific intervals, triggering ADC conversions, or controlling timing-critical operations. - Prescaler and Period: Set the appropriate prescaler and period values for Timer0 to achieve the desired timing resolution and interval.
- Interrupts: Configure Timer0 to generate interrupts when the specified timing interval is reached. Utilize Timer0 interrupts to synchronize tasks, trigger ADC conversions, or perform other time-dependent operations.

4. ADCISR (ADC Interrupt Service Routine):
- Interrupt Handling: Implement the ADCISR as an interrupt service routine to handle the ADC interrupt triggered by the completion of ADC conversions. The ADCISR should be properly defined and linked to the corresponding interrupt vector.

- Data Processing: Inside the ADCISR, retrieve the ADC conversion result (ADCRESULT0) from the ADC registers and perform the necessary data processing or analysis. In this case, compare the ADC result (j) with the specified threshold values (1700 and 2500) to control the MOSFET switch through GPIO16.
- Clearing Flags: Clear the ADC interrupt flag (ADCINT1) to acknowledge and prepare for the next ADC conversion. Additionally, clear the interrupt acknowledge bit (PIEACK) to clear the interrupt at the CPU level.

By effectively planning and managing these resources, developers can ensure the proper functioning of the ADC, GPIO, CPU Timer0, and ADCISR components in the solenoid control circuit. Proper configuration, utilization, and synchronization of these resources are essential to achieve accurate current control and efficient operation of the overall system.

CODE

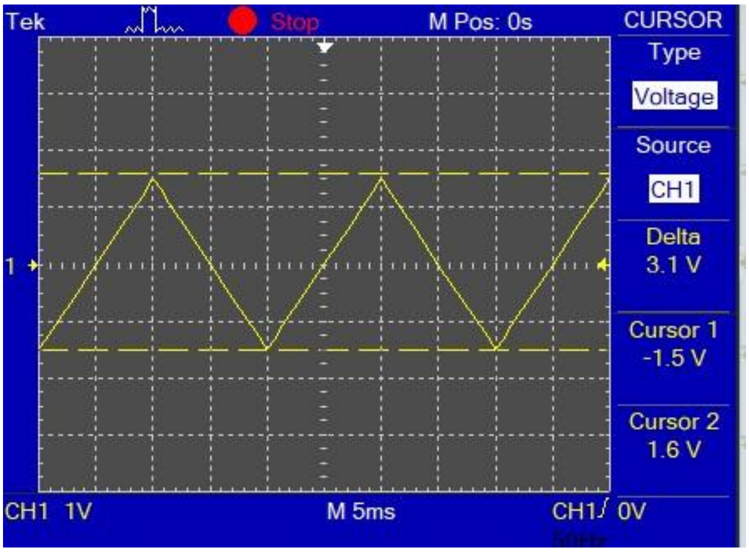
```
#include
"F28x_Project.h"      void
SysControl(void);     void
Adc(void);            void
Timer0(void);         void
GpioInit(void);
__interrupt void Adcisr(void);
int i,j;
void main(void)
{
   DINT;//disable the interrupt

   SysControl();//function call
   Timer0();
   Adc();
   GpioInit();
   EALLOW;
   PieCtrlRegs.PIECTRL.bit.ENPIE=1;//enable the interrupt in PIE stage
   PieCtrlRegs.PIEIER1.bit.INTx1=1;//enable the ADC interrupt at PIE level
   PieVectTable.ADCA1_INT= &Adcisr;//write the address of the function,peripheral
level
   PieCtrlRegs.PIEACK.bit.ACK1=1;//clear the acknowledge bit
   EDIS;
   IER=0x0001;//enable the interrupt at CPU stage
EINT;//enable the global interrupt     while(1);
}
void SysControl(void)
{
   EALLOW;
   ClkCfgRegs.CLKSRCCTL1.bit.OSCCLKSRCSEL=00;//select the clock source
ClkCfgRegs.SYSPLLCTL1.bit.PLLCLKEN=0;
   for(i=0;i<=120;i++);
   ClkCfgRegs.SYSCLKDIVSEL.bit.PLLSYSCLKDIV=0;//clear this bit to ensure
fastest PLL configuration     for(i=0;i<5;i++)
   {   ClkCfgRegs.SYSPLLCTL1.bit.PLLEN=0;//lock PLL
```

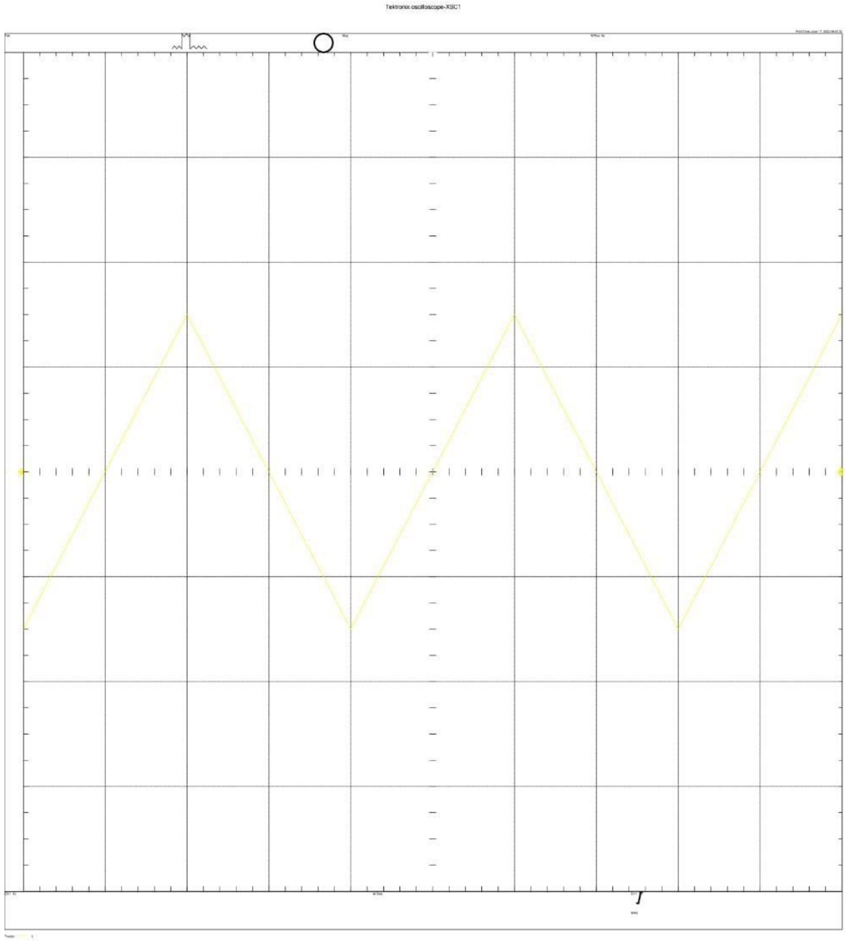
```
      ClkCfgRegs.SYSPLLMULT.bit.FMULT=00;//no fractional mult
ClkCfgRegs.SYSPLLMULT.bit.IMULT=16;//10Mhz PLL raw clock
while(ClkCfgRegs.SYSPLLSTS.bit.LOCKS!=1);
   }

   ClkCfgRegs.SYSCLKDIVSEL.bit.PLLSYSCLKDIV=1;
ClkCfgRegs.SYSPLLCTL1.bit.PLLCLKEN=1;
   ClkCfgRegs.SYSCLKDIVSEL.bit.PLLSYSCLKDIV=0;
   ClkCfgRegs.LOSPCP.bit.LSPCLKDIV=1;
   //set the low speed clock
   EDIS;

}
void Timer0(void)
{   EALLOW;
   CpuSysRegs.PCLKCR0.bit.CPUTIMER0=1;//turn on the Timer0 module clock
   CpuTimer0Regs.TCR.bit.TSS=1;//stop the timer so as to set the initial condition
CpuTimer0Regs.TPR.bit.PSC=100;//set the prescalar so as to obtain lesser frequencies
generated by system clock
   CpuTimer0Regs.PRD.all=50000;//set the period register to desired period
   CpuTimer0Regs.TCR.bit.TIF=1;//clear overflow flag
   CpuTimer0Regs.TCR.bit.TIE=1;//enable the interrupt
   CpuTimer0Regs.TCR.bit.FREE=1;//set to free run mode
   CpuTimer0Regs.TCR.bit.TRB=1;//reload register used to load the any changed value
   CpuTimer0Regs.TCR.bit.TSS=0;//start the timer since all the initial values are set
   EDIS;
}

void Adc(void)
{
   EALLOW;
   CpuSysRegs.PCLKCR13.bit.ADC_A=1; //Enable the clock for the ADC
   AdcaRegs.ADCCTL2.bit.PRESCALE=0x03; //Set the ADC clock frequency
   AdcaRegs.ADCCTL2.bit.SIGNALMODE=0;
   AdcaRegs.ADCCTL2.bit.RESOLUTION=0;//Resolution,Signal mode,Ref Hi,Ref
loww
   AdcaRegs.ADCCTL1.bit.ADCPWDNZ=1;//Power up for ADC
   AdcaRegs.ADCSOC0CTL.bit.CHSEL=2;
```

```
   AdcaRegs.ADCSOC0CTL.bit.TRIGSEL=01;
   AdcaRegs.ADCSOC0CTL.bit.ACQPS=83;//Setting up SOC's
   AdcaRegs.ADCCTL1.bit.INTPULSEPOS=1;//Interrupt position select(at the end of
conversion there will  be interrupt)
   AdcaRegs.ADCINTSEL1N2.bit.INT1SEL=0;//if i have selected more than 1 channel
then here i should put the last channel that i have chosen (which has the highest number)
AdcaRegs.ADCINTSEL1N2.bit.INT1E=1;//Select and Enable the interrupt
   EDIS;

}
void Adcisr(void)
{
   EALLOW;
   CpuTimer0Regs.TCR.bit.TIF=1;//clear the overflow flag
   j=AdcaResultRegs.ADCRESULT0&0X0FFF;//reading the ADC that is from
peripheral to ADC
   //or
   //int ResulrArray[0]=AdcaResultRegs.ADCRESULT0;

   if(j<1700)
      GpioDataRegs.GPASET.bit.GPIO16=1;
else if(j>2500)
      GpioDataRegs.GPACLEAR.bit.GPIO16=1;

   AdcaRegs.ADCINTFLGCLR.bit.ADCINT1=1;//ADC interrupt flag to be cleared
   PieCtrlRegs.PIEACK.bit.ACK1=1;//clear the acknowledge bit
   EDIS;

}
void GpioInit(void)
{
   EALLOW;//protected
   GpioCtrlRegs.GPADIR.bit.GPIO16=1;//make GPIO18 as OUTput pin
   GpioCtrlRegs.GPAGMUX2.bit.GPIO16=00;
   GpioCtrlRegs.GPAMUX2.bit.GPIO16=00;
GpioCtrlRegs.GPAPUD.bit.GPIO16=00;

   EDIS;
```

}

OUTPUT WAVEFORM





REFERENCES

[1] M H. RASHID, Power electronics circuits, devices and applications, 4th edition, Pearson Publications.

[2] DAVID A BEL, Electronic devices and circuits 5th edition, Oxford Publications

[3] https://datasheetspdf.com/pdf-file/806659/Vishay/IRF540/1

[4]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9923626&isnumber=10108397

[5] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=316396&isnumber=7625