Lab 4: Control of Duty Cycle via PWM on the TI F28335

Using MATLAB and Simulink for Pulse Width Modulation

Work done by :

CHERIF Enzo

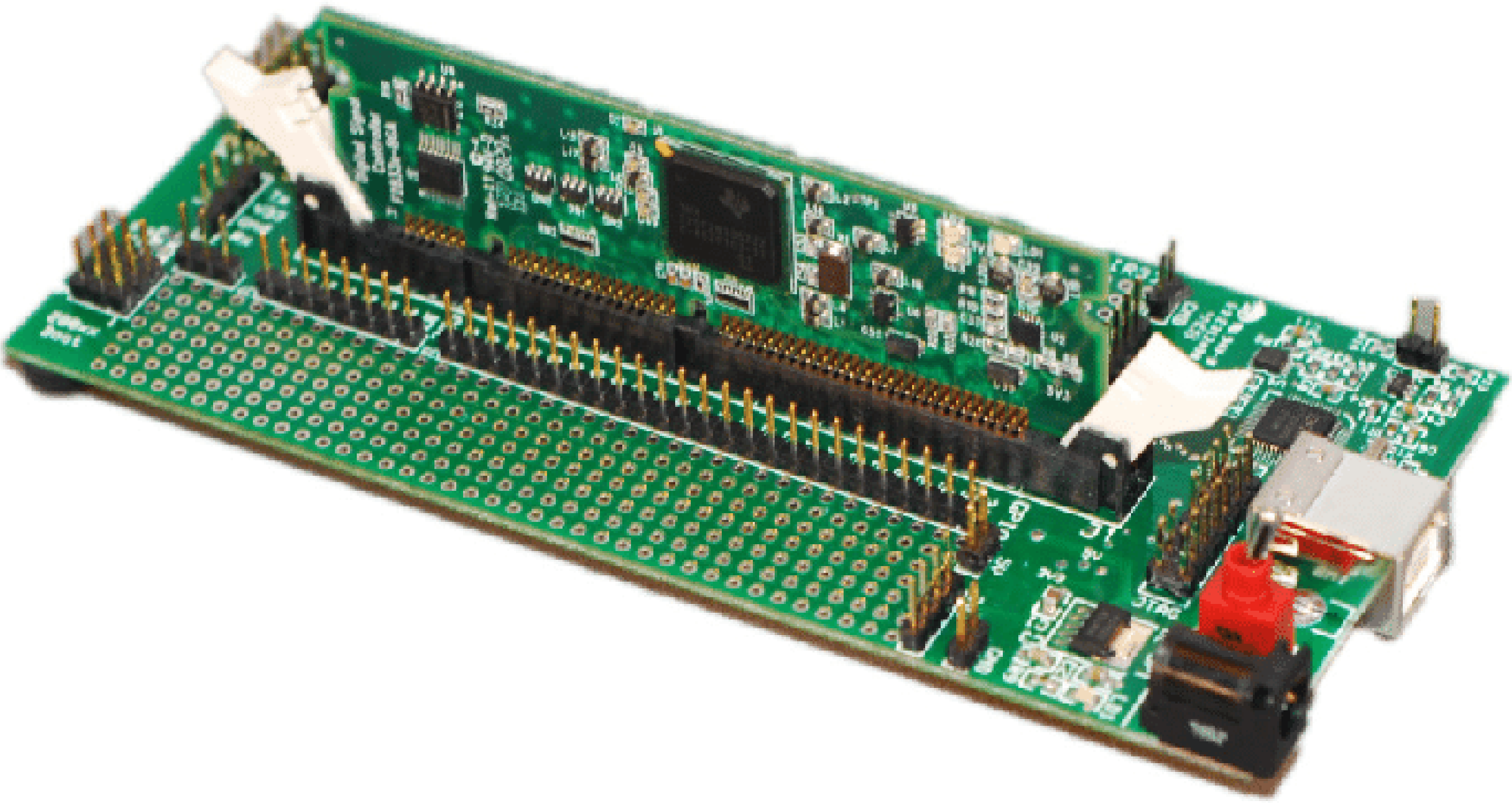
Students at Seatech in Class of 2025

2A, SYSMER track

As part of the course :

Programming on electronic card

May 2024



Contents

1 Introduction **2**
1.1 Objective of the Lab: PWM Signal Modification by Duty Cycle . . . 2
1.2 Presentation of Tools to Use: ADC, Potentiometer . . . 2

2 Theoretical Fundamentals **3**
2.1 Review of Basic Concepts on ADC and Duty Cycle . . . 3
2.2 Explanation of Reference Values for 12-bit ADC . . . 4

3 Required Hardware and Software Configuration **5**

4 Manipulation of Duty Cycle **6**

5 Experimental Procedure **9**

6 Analysis and Interpretation of Results **11**

1 Introduction

1.1 Objective of the Lab: PWM Signal Modification by Duty Cycle

The main goal of this practical work is to allow students to understand and manipulate the pulse width modulation (PWM) signal by dynamically adjusting the duty cycle. The duty cycle, which represents the fraction of time during which a signal is in the high state in a complete cycle, is a crucial parameter in many electronic and control applications, particularly for controlling the power delivered to a device.

Specific Goals

1. **Practical Manipulation of the Duty Cycle**:
   - Use a potentiometer connected to an analog-to-digital converter (ADC) to adjust the duty cycle of the PWM signal in real time.
   - Observe how changes in the duty cycle directly affect the system's outputs, thereby allowing for a deeper understanding of the relationship between analog inputs and digital outputs in embedded systems.
2. **Application in Simulation and Real Time**:
   - Integrate theoretical concepts with practice using simulation tools such as MATLAB/Simulink to model the behavior of the PWM signal with different duty cycle values.
   - Implement the system on the TI F28335 board, allowing students to see the effect of their adjustments in real time.

Skills to Acquire

- **Technical Analysis**:
  - Develop the ability to analyze and understand electronic diagrams and digital signals.
  - Enhance the understanding of the interaction between hardware and software in embedded systems.
- **Development of Critical Thinking**:
  - Encourage experimentation and active adjustment of parameters to see their immediate impacts, thus cultivating a critical and experimental approach to problem solving.
- **Engineering Skills**:
  - Strengthen skills in designing and optimizing systems through the use of standard electronic components and their integration into more complex applications.

This practical session is designed not only to reinforce students' technical knowledge about PWM but also to improve their ability to control and optimize electronic systems in practical contexts. Ultimately, students will be better prepared to design and implement sophisticated control systems in their future engineering projects.

1.2 Presentation of Tools to Use: ADC, Potentiometer

For this practical work, two essential components will be used to enable the manipulation of the PWM signal by the duty cycle: the analog-to-digital converter (ADC) and the potentiometer. Here is an introduction to each tool and its role in the experiment.

Analog-to-Digital Converter (ADC)

1. **Function of the ADC**:
   - The ADC is a device that converts an analog signal (which varies continuously in amplitude) into a digital signal (which has discrete values). For this practical work, the ADC of the TI F28335 board will be used to read the variable voltage provided by the potentiometer.
   - The ADC on the TI F28335 board has a resolution of 12 bits, which means it can represent an analog input by one of $4096 = 2^{12}$ different values, thus offering sufficient precision for most control applications.
2. **Importance for the Practical Work**:

- By measuring the voltage across the potentiometer, the ADC transforms this value into a numerical number that will be used to adjust the duty cycle of the PWM. This allows for precise and easily quantifiable manipulation of the PWM signal.

Potentiometer

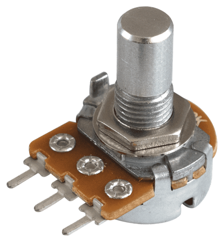


Figure 1: View of a potentiometer used to adjust the duty cycle in PWM systems.

1. **Function of the Potentiometer**:
   - A potentiometer is a form of variable resistor that can be manually adjusted to change the resistance across a circuit. In this practical work, it acts as a voltage divider, where its position adjusts the output voltage between 0 and 3.3 volts.
   - By turning the knob of the potentiometer, students can vary the output voltage that is read by the ADC.
2. **Importance for the Practical Work**:
   - The potentiometer allows students to actively modify the input voltage to the ADC, which translates into a change in the PWM duty cycle. This gives students direct and intuitive control over the PWM signal and clearly illustrates the concepts of analog-to-digital conversion and signal modulation.

Practical Application

- Students will connect the potentiometer so that its output is linked to the ADC input on the TI F28335 board. By adjusting the potentiometer, they will observe how the duty cycle of the PWM signal changes in response to the analog input, thus visualizing the direct effect of their manipulation on the system's behavior.

This section of the practical work is crucial as it allows students to see how basic electronic components can be used in more complex control applications and how changes in the physical world (turning a potentiometer) are converted into digital actions (modifying the duty cycle) that have measurable practical effects.

2 Theoretical Fundamentals

2.1 Review of Basic Concepts on ADC and Duty Cycle

To fully understand the practical manipulations in this lab, it is essential to master the basic theoretical concepts concerning the analog-to-digital converter (ADC) and the duty cycle. These two elements play crucial roles in pulse width modulation (PWM) and in controlling various electronic devices.

1. **Analog-to-Digital Converter (ADC)**:
   - **Definition and Function**: The ADC is a device that converts a continuous (analog) signal into a discrete (digital) signal. The analog signal can vary unrestrictedly across a range of values, whereas the digital signal can only take specific values.
   - **Resolution**: The resolution of the ADC indicates the number of discrete values it can produce over the measured voltage range. For example, a 12-bit ADC, like the one on the TI F28335 board, can produce 2^{12}, or 4096 different values. The resolution directly influences the accuracy with which an analog signal is digitally represented.
   - **Importance in PWM**: In the context of PWM, the ADC is used to measure a variable voltage (such as that from a potentiometer) and then dynamically adjust the duty cycle of the PWM signal based on this measurement.

2. **Duty Cycle**:
   - **Definition**: The duty cycle is a key feature of PWM that determines the proportion of time the signal is at a high level (ON) relative to the total cycle period. It is generally expressed as a percentage.
   - **Calculation**:
   $$\text{Duty Cycle (\%)} = \left(\frac{T_{on}}{T}\right) \times 100$$
   where T_{on} is the time during which the signal is high and T is the total cycle period.
   - **Practical Implications**: Modifying the duty cycle directly affects the average amount of power delivered to a device. For example, increasing the duty cycle on a motor results in an increase in its speed, while on an LED, it increases the brightness.

Application of Concepts

These theoretical concepts are applied in the lab in such a way that students can not only understand but also experiment with the relationship between the voltage measured by the ADC and its effect on the PWM duty cycle. By adjusting the potentiometer, students alter the input voltage, which is then converted by the ADC into a numerical value used to adjust the duty cycle. This practical experience reinforces the understanding of theory by showing how analog adjustments influence digital behavior and, by extension, the physical devices controlled by PWM.

2.2 Explanation of Reference Values for 12-bit ADC

To effectively utilize the analog-to-digital converter (ADC) in your PWM applications, it is crucial to understand the reference values used by a 12-bit ADC. These values determine the resolution and accuracy of the measurements that the ADC can provide.

Understanding Reference Values

1. **ADC Resolution**:
   - A 12-bit ADC has a resolution that allows it to divide the input voltage range into 2^{12} distinct levels, or 4096 different levels. Each level represents the smallest voltage change that the ADC can detect.
2. **Reference Voltage Range (V_{ref})**:
   - V_{ref} is the maximum voltage that the ADC can measure. For the TI F28335 board, this voltage is typically 3.3 volts. This value is crucial as it determines the voltage corresponding to the highest digital level that the ADC can output.
   - For example, if $V_{ref} = 3.3V$, the maximum level of 4095 (since counting starts at zero) corresponds to 3.3 volts.

Understanding Binary Values and Their Conversion

To deepen the understanding of how a 12-bit ADC processes signals, it is useful to consider the binary values that these bits can represent. Each bit can be 0 or 1, which gives for the first and last level:

- **First Binary Value**: $0000\ 0000\ 0000_2$
- **Last Binary Value**: $1111\ 1111\ 1111_2$

These binary representations correspond respectively to the smallest and largest values that the ADC can convert into a numerical value, ranging from 0 for the minimum voltage to 4095 for the maximum voltage.

Simplified Example on 4 Bits

To make this idea more accessible, consider a simpler 4-bit ADC. The possible binary values range from:

- **Minimum**: $0000_2 \rightarrow 0_{10}$
- **Maximum**: $1111_2 \rightarrow 2^4 - 1 = 15_{10}$

This 4-bit system illustrates the basic concept that each additional bit doubles the number of possible values, thereby improving the resolution of the ADC. For example:

- **With 4 bits**, the range of possible values is from 0 to 15.
- **With 8 bits**, this range extends from 0 to $2^8 - 1 = 255$.
- **With 12 bits**, the range reaches $2^{12} - 1 = 4095$, as on our TI F28335 board.

This means that for a 12-bit ADC with a reference voltage of 3.3 volts, each increment of 1 in the numerical value represents a very small but precise increase in the measured voltage, allowing for very fine and detailed measurements. This high resolution is crucial for applications requiring great precision, such as precise duty cycle control in PWM systems.

Converting Voltage to Numerical Values

To convert an analog voltage into a numerical value, the ADC uses the following formula:

$$\text{Numerical Value} = \left(\frac{\text{Measured Voltage}}{V_{ref}}\right) \times (2^{12} - 1)$$

where $2^{12} - 1 = 4095$ is the maximum level for a 12-bit ADC.

Practical Examples

- **Mid-Scale Measurement**:
  - If the measured voltage is half of V_{ref}, or 1.65 volts, the calculated numerical value would be:

    $$\text{Numerical Value} = \left(\frac{1.65V}{3.3V}\right) \times 4095 \approx 2048$$

    This result indicates that for a voltage of 1.65 volts, the ADC produces a numerical value of approximately 2048, which represents exactly half of the total range.
- **Calculating Voltage for a Given Numerical Value**:
  - Conversely, if the numerical value produced by the ADC is known, the corresponding voltage can be calculated using the reversed formula:

    $$\text{Measured Voltage} = \left(\frac{\text{Numerical Value}}{4095}\right) \times 3.3V$$

Importance for the Practical Work

This information is essential for students as it allows them to understand how variations in analog inputs are translated into precise numerical values. By adjusting the potentiometer, which varies the input voltage between 0 and 3.3V, students can see how these changes directly affect the numerical value produced by the ADC, and thus how they influence the duty cycle of the PWM signal. This understanding is crucial for effectively manipulating PWM-based control systems.

3 Required Hardware and Software Configuration

List of Required Materials and Components

To complete this lab session, you will need the following items:

1. **TI 28335 Board**: A development board equipped with a 12-bit ADC and capable of handling PWM signals.
2. **Potentiometer**: A 10k potentiometer will be used to adjust the input voltage to the ADC.
3. **Connection Cables**: Jumper wires are necessary to make connections between the potentiometer, the TI 28335 board, and the power source.
4. **3.3 V Power Source**: Used to power the potentiometer.
5. **Computer with MATLAB/Simulink**: Necessary for programming and simulating the circuit behavior through the interface of the TI 28335 board.
6. **Voltmeter** (optional): To verify the voltage at different points of the circuit.

Circuit Setup Description

The circuit setup is relatively straightforward; there's no need to make the connections just yet, we will return to that later:

1. **Connecting the Potentiometer**: The potentiometer should be connected with its central terminal to the ADC input of the TI 28335 board. The other two terminals should be connected respectively to the 3.3 V power source and the ground (GND).

2. **Connection to the TI 28335 Board**: The central terminal of the potentiometer (the wiper) is connected to the ADC0 input of the board, allowing the ADC to read the variable voltage adjusted by the potentiometer.

3. **Power Supply**: Ensure that the TI 28335 board is correctly powered according to the manufacturer's specifications, often through a USB connection to a computer or an external power source.

To successfully complete this lab session, it is essential that MATLAB and Simulink are properly installed. If this is not the case, I invite you to follow the instructions from Lab 1.

Importing and Adjusting the ADC-PWM Example

Once MATLAB and Simulink are configured, you can proceed to import and adjust the ADC-PWM example. This process will help you understand how analog inputs are converted into controllable digital signals, which are essential for adjusting the duty cycle in pulse width modulation (PWM) systems.

1. **Opening the ADC-PWM Example**:

   - **Launching Simulink**: Start Simulink from MATLAB by clicking on the Simulink icon in the "Home" tab of the MATLAB interface.
   - **Reusing the Preconfigured Model**: In Lab 1, an ADC-PWM example was set up to illustrate the conversion of analog signals into PWM commands. If you have saved this model, you can directly reopen it to continue from there.
   - **Accessing Simulink Examples**: If the preconfigured model is not available, navigate through the Simulink example library. Use the search function to type "ADC PWM" and select the first example.
   - **Selecting the Appropriate Model**: Open this example that meets the requirements of the lab. These models are useful because they already contain the necessary blocks for manipulating the ADC and generating PWM signals. Ensure to configure the model as in Lab 1.
   - **Desired Model**: The model we would like to have is as follows. If you do not have exactly the same thing, it is not a problem, but you should still have roughly this basis:

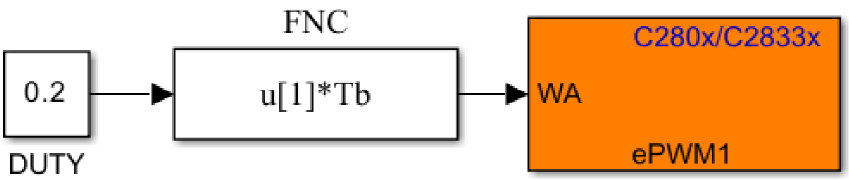


Figure 2: Basic schematic of the ADC-PWM on MATLAB

These steps will help you prepare your software environment and adjust the necessary examples for the lab, ensuring a successful practical session with the TI 28335 board.

4 Manipulation of Duty Cycle

Dynamic Adjustment of the Duty Cycle via the ADC

Using the potentiometer to control the duty cycle is accomplished through the ADC on the TI 28335 board. Here's how it works in a practical and interactive manner:

1. **Voltage Measurement with a Voltmeter**:

- To validate the proper functioning of the setup, connect a voltmeter at the appropriate points. Place the positive (red) probe of the voltmeter on the ADC0 input, where the potentiometer sends the signal, and the negative (black) probe on the GND (ground).

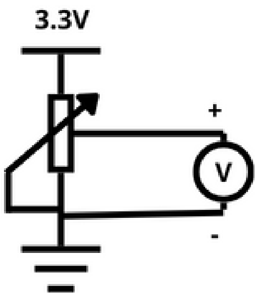


Figure 3: Connecting the voltmeter to the potentiometer.

- By adjusting the potentiometer, observe the variation in voltage on the voltmeter. This voltage should vary from 0 to 3.3V, which is ideal for our application.

2. **Observation of Potentiometer Functioning**:
   - Turn on the TI 28335 board and start turning the knob of the potentiometer. You should see the voltage measured by the voltmeter vary accordingly. This allows you to verify if the potentiometer is functioning correctly and ensures that the board reacts well to changes.

3. **Relationship Between Voltage and Duty Cycle**:
   - The change in voltage measured by the ADC is converted into a numerical value (from 0 to 4095 for a 12-bit ADC). This numerical value is then used to adjust the duty cycle of the PWM signal. The duty cycle value is calculated from the ADC output, which reads the variable voltage adjusted by the potentiometer. For example, a maximum voltage of 3.3V corresponds to an ADC value of 4095, which should yield a duty cycle of 100%.

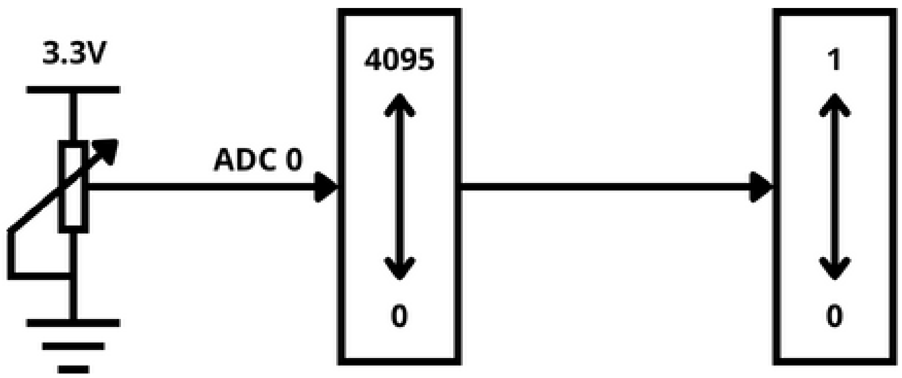


Figure 4: Description of the Duty Cycle parameter modification via the ADC.

Re-adjusting the MATLAB/Simulink Model

To dynamically adjust the duty cycle based on ADC readings, follow these steps in MATLAB/Simulink:

1. **Modify the Duty Cycle Block**:
   - Initially, the duty cycle value may be fixed or configured for static tests (see the figure above Figure 2). Your goal is to make it dynamic, controllable via the ADC signal.
   - Replace the static duty cycle block with a link to the ADC output, adjusted by an appropriate gain factor to convert the range of 0-4095 to a ratio of 0 to 1 (0% to 100% duty cycle). For now, the value of this gain is not accurate—it's just to show what the model should look like.

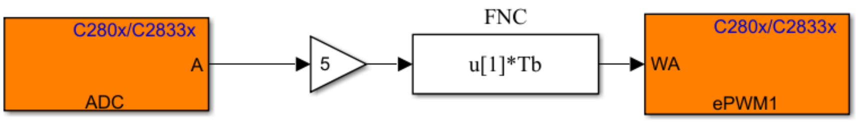


Figure 5: Simulink model illustrating duty cycle adjustment using a gain factor on the ADC output.

- To ensure your setup is correct, you should see this in your Simulink model. If you cannot find the necessary block, you have two options: reopen a previously used example or search in the Simulink library using the board name to find the appropriate block. Don't worry if the block you find does not have the same color as expected; the color of the block does not affect its functionalities or characteristics.

2. **Determine the Gain Value**:

   - In the duty cycle control model via an ADC signal, the gain necessary to transform the ADC output into a ratio suitable for the duty cycle is crucial. The figure below shows a simplified diagram of this model. Block A denotes the FNC and PWM block.

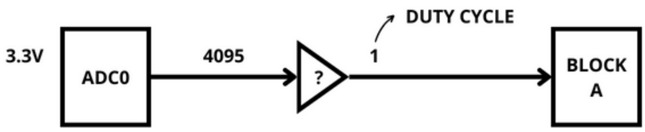


Figure 6: Diagram of calculating the duty cycle from the ADC0 output.

For a maximum input voltage of 3.3V, corresponding to an ADC reading of 4095, the gain is determined by the following relationship:

$$4095 \times \text{gain} = 1 \Rightarrow \text{gain} = \frac{1}{4095} \quad (1)$$

The solution to this simple equation reveals that the gain must be $\frac{1}{4095}$.

   - Adjust this gain in your model to ensure that the conversion is accurate and correctly reflects incoming voltage variations.

3. **ADC Configuration**:

   - Thus, for testing, here are all the remaining steps we have to do: ensure that the ADC block is well configured. To do this, double-click on it. This will open the settings window, and put the same parameters as shown below:

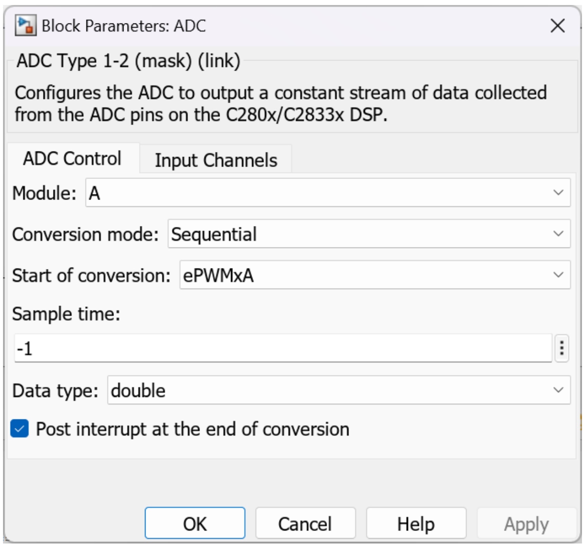


Figure 7: ADC control settings window.

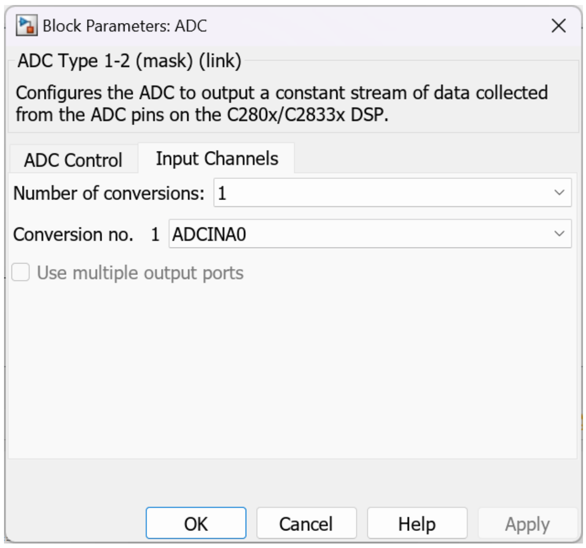


Figure 8: ADC input channels settings window.

- You may not take the same port parameters and choose a port other than ADC0, but I advise using this one because in the rest of the lab, it is the one we use, and to simplify your task, all connections will be explained with ADC0. Be sure to choose "Data Type = Double". In programming, the ‘double‘ type represents a floating-point number, which allows for great precision and the ability to represent very small or very large values, ideal for complex calculations or precise measurements. On the other hand, ‘uint16‘ is an unsigned integer data type that can store numbers from 0 to 65535, used for data that does not require fractions and where memory conservation is more critical.

These modifications allow for a system where the duty cycle of the PWM signal is directly controlled by the adjusted voltage via the potentiometer, offering an intuitive and effective method to manipulate the behavior of your application in real time.

5 Experimental Procedure

Hardware Setup and Connections

1. **Preparing the Hardware**:
   - Gather the TI 28335 board, the necessary cables for connecting the potentiometer and the oscilloscope, and a PC to establish communication with the board.
2. **Connecting the Potentiometer**:
   - Connect the leftmost wire of the potentiometer to the 3V3 pin of the board. Connect the middle wire to ADC0, and the right wire to the GND of the board.

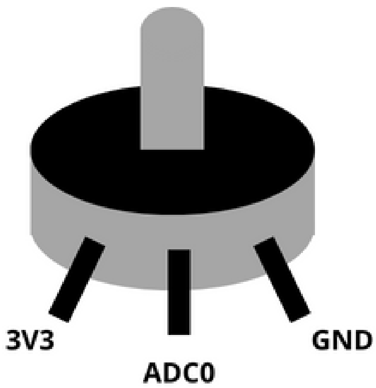


Figure 9: Potentiometer Connection

3. **Connecting the Oscilloscope**:
   - Perform the same connections as in Lab 1 for the oscilloscope: connect one probe to the GPI0 00 pin (to observe the PWM signal) and the other probe to GND.

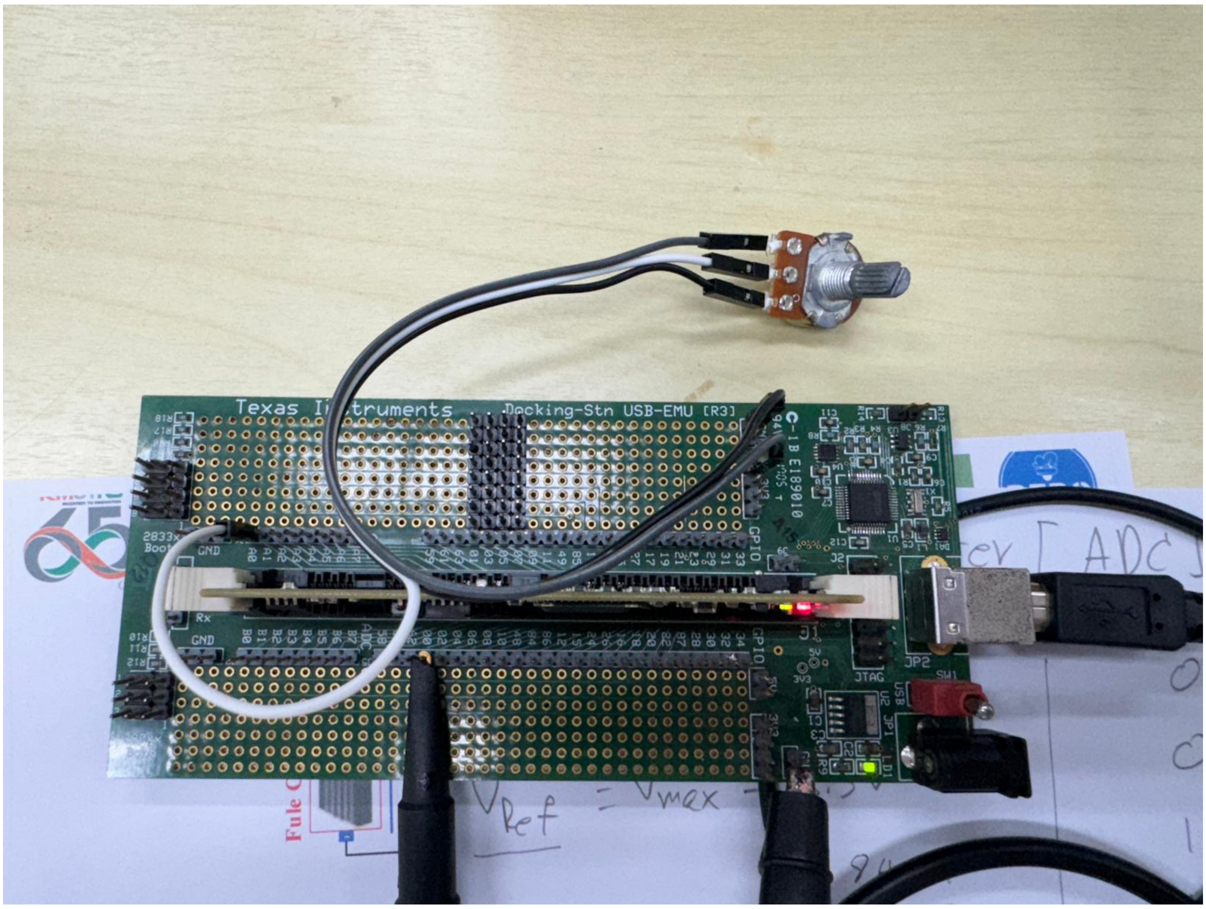


Figure 10: Oscilloscope Connection

4. **Connecting to the PC**:
   - Using a USB cable, connect the TI 28335 board to your PC. Ensure that the board is correctly detected by the system.

System Testing

1. **Verifying the MATLAB/Simulink Model**:
   - Before proceeding to send the build, review your diagram in MATLAB to ensure it is correctly configured according to the lab requirements.
2. **Compiling the Model**:
   - Use the Ctrl+B shortcut to compile and download the program to the board. Compilation may take some time, especially if you are using a PC with modest specifications.
3. **Positioning the Potentiometer**:
   - Set the potentiometer to a mid-range position, to avoid being at the ends. This will allow starting the tests with a signal that is neither too high nor too low.
4. **Observing with the Oscilloscope**:
   - Turn on the oscilloscope and observe the signal. You should see a PWM signal whose pulse width varies according to the position of the potentiometer.

Analysis of Results

1. **Interpreting Data**:
   - As you turn the potentiometer, you should observe a corresponding variation in the signal on the oscilloscope, indicating that the duty cycle of the PWM signal is indeed changing in real time.

2. **Fine Tuning**:
   - If the observed signal does not match what is expected, you may need to return to Simulink to adjust the settings, especially the gain applied after the ADC to properly match the duty cycle range.

Following this procedure, you will be able to see the concrete effect of adjusting the potentiometer on the duty cycle of the PWM signal, and understand how analog signals can be used to control digital systems.

6 Analysis and Interpretation of Results

Analysis and Interpretation of Results

Guide on Analyzing the Obtained Data

Once the tests have been conducted with the oscilloscope, it's time to analyze the data:

1. **Documenting Observations**:
   - Record the voltage displayed by the oscilloscope for each position of the potentiometer. You should have a series of values indicating the voltage of the PWM signal at different duty cycles.
2. **Correlation Between Potentiometer and Duty Cycle**:
   - Establish a correspondence between the position of the potentiometer (which you can quantify in terms of rotation degrees or percentage) and the observed duty cycle value.
3. **Establishing a Response Curve**:
   - If possible, create a graph showing the relationship between the position of the potentiometer and the duty cycle of the PWM signal. This can be done in post-processing if the oscilloscope does not directly allow it. You should have a curve similar to this:

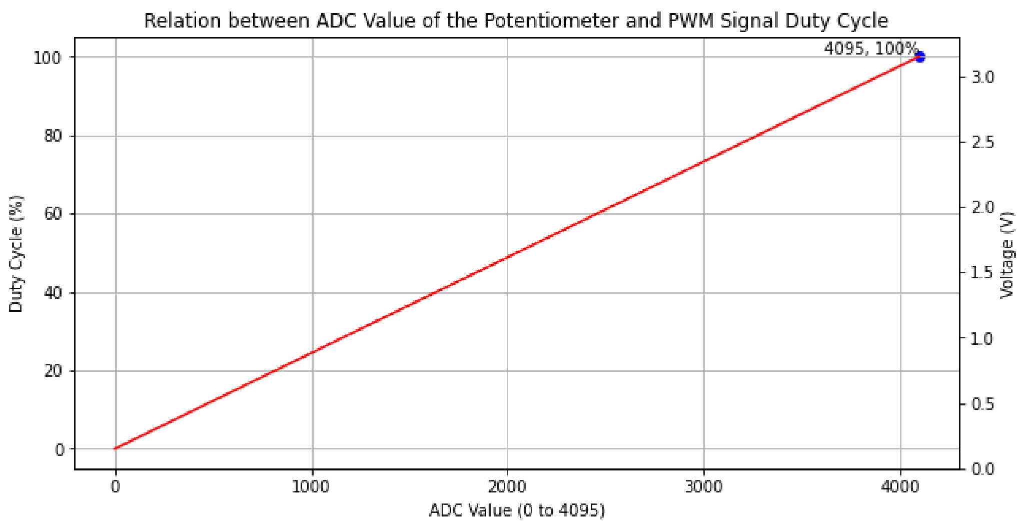


Figure 11: Relationship between the ADC value of the potentiometer and the duty cycle of the PWM signal.

Reflection Questions and Explanations

To deepen the understanding of the concepts covered in this lab session and test your knowledge, here are some reflection questions with their explanations:

Question 1: What is the effect of increasing the potentiometer voltage on the duty cycle of the PWM signal?

Answer: ...

Question 2: Why is it important to verify the linearity of the duty cycle response relative to the position of the potentiometer?

Answer: ...

Question 3: How would you interpret a sudden change in the duty cycle that does not correspond to adjustments of the potentiometer?

Answer: ...

Question 4: If the measured duty cycle does not change despite rotating the potentiometer, what could be the possible causes?

Answer: ...

These questions are designed to encourage critical thinking about the interactions between electronic components and software modeling, as well as the importance of systematic verification in implementing control systems. By carefully analyzing the data and understanding its significance, you can draw accurate conclusions about system behavior and make informed decisions for future projects or improvements.

This lab session explored how to manipulate the duty cycle of a PWM signal via a potentiometer and ADC on the TI 28335 board, demonstrating the interaction between analog inputs and digital controls. Practical testing and analysis have reinforced the understanding of linearity and system responses, providing valuable experience in troubleshooting and parameter adjustment in control systems. This hands-on approach is crucial for mastering skills in electronic and mechanical engineering.

Explanation

Question 1: What is the effect of increasing the potentiometer voltage on the duty cycle of the PWM signal?

Answer: Increasing the potentiometer voltage increases the value read by the ADC, which in turn increases the duty cycle of the PWM signal. This means that the signal stays at a high level longer during each cycle. This is because the duty cycle is directly proportional to the input voltage converted by the ADC.

Question 2: Why is it important to verify the linearity of the duty cycle response relative to the position of the potentiometer?

Answer: Verifying linearity helps to ensure that the system behaves in a predictable and consistent manner. In industrial applications, a nonlinear response can lead to unpredictable and potentially dangerous performance. Identifying linearity allows for system adjustments to achieve desired responses or to diagnose potential issues with components such as the potentiometer or ADC configurations.

Question 3: How would you interpret a sudden change in the duty cycle that does not correspond to adjustments of the potentiometer?

Answer: A sudden and unexpected change in the duty cycle may indicate an issue either with the potentiometer, the circuit (such as a poor connection or a short circuit), or the ADC itself. It might also suggest a problem in the Simulink model code or configuration, such as an error in gain calculation or a misinterpretation of ADC values. It is crucial to troubleshoot these aspects to ensure the proper functioning of the system.

Question 4: If the measured duty cycle does not change despite rotating the potentiometer, what could be the possible causes?

Answer: If the duty cycle remains constant despite changes in the position of the potentiometer, several factors might be at play:

- The potentiometer is faulty or poorly connected.
- The ADC is not reading input values correctly, possibly due to incorrect configuration or a hardware issue.
- There might be a problem in the transmission of data between the ADC and the signal processing block in Simulink, necessitating a check of the wiring and model settings.