

Design (E) 314

Preliminary Report

PV System Efficiency Monitor

Author: Student Number:

Dgibrilly M. Mutabazi 23765518

[01/04/2024)]

**Plagiaatverklaring / Plagiarism Declaration**

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

*Plagiarism is the use of ideas, material and other intellectual property of another’s work and to present is as my own.*

1. Ek erken dat die pleeg van plagiaat ’n strafbare oortreding is aangesien dit ’n vorm van diefstal is.

*I agree that plagiarism is a punishable offence because it constitutes theft.*

1. Ek verstaan ook dat direkte vertalings plagiaat is.

*I also understand that direct translations are plagiarism.*

1. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

*Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.*

1. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of ’n ander module/werkstuk nie.

*I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.*

|  |  |  |
| --- | --- | --- |
|  |  | 23765518 |
| Handtekening / *Signature* |  | Studentenommer / *Student number* |
| D.M Mutabazi |  | 20/05/2024 |
| Voorletters en van / *Initials and surname* |  | Datum / *Date* |

**Abstract (paraphase)**

This report details the design, development, and testing of a multi-functional device tailored to monitor and report PV module performance under varying environmental conditions. User requirements encompassing power measurements, environmental data acquisition, calibration procedures, and real-time communication protocols guided the development process. Key features include precise PV module measurements, ambient environment monitoring, real-time data retrieval, and an intuitive user interface via UART commands and push-buttons. The device's potential impact lies in its ability to empower PV system owners with actionable insights, enhancing system efficiency, and prolonging equipment lifespan. Experimental results validate the device's functionality and reliability in real-world PV system applications, highlighting its practical significance in sustainable energy management.

**Table of contents**

**Task 1: Hardware Design details……………………………………….4**

**LMT01 Sensore………………………………………………………………….4**

**Top push button………………………………………………………………..5**

**LED circuit………………………………………………………..……………….5**

**Task 2: Software Design Details………………………………………..6**

**LMT01 description……………………………………………………………6**

**Software Design………………………………………………….………….6-7**

**Task 3: Testing of system to verify performance……………....8**

**LMT01…………………………………………………………………….…………8**

**Top push button - active low……………………………….…………….8**

**LED circuit…………………………………….…………………………………..9**

**LMT01 Functionality: reaction to changing temperature…..9**

**Referenes…………………………………………………………………………10**

**List of Figures**

**List of Tables**

**1 Introduction (paraphrase)**

As the adoption of solar photovoltaic (PV) systems grows in South Africa, so does the need for efficient monitoring and maintenance strategies. Dirty or soiled PV modules can significantly reduce power output, impacting the return on investment for system owners. Recognizing this challenge, our project focuses on developing a PV System Efficiency Monitor—a device aimed at helping PV system owners determine the extent of power loss due to dirt and soil accumulation, thereby optimizing cleaning schedules and maximizing energy production.

**2 System description**

**2.1 STM32F303RE microcontroller board**

**2.2 Power Supply**

**2.3 UART Communication**

**2.4 Push – buttons**

**2.5 LEDs**

**2.6 LCD**

**2.7 ADC**

The STM32F411RE NUCLEO has 1 ADC peripheral with 16 multiplexed channels. The system measure 4 single input channels for 1 (Voltage+) PV voltage, 2) Voltage – (voltage after the current sense circuit, to calculate the PV current), 3) LM235 ambient temperature sensor and 4) SFH203 photodiode output. The ADC channels inputs are at PB1, PC5, PA0 and PC4 respectively.

**2.8 Op-Amp**

**2.9 Photodiode**

**2.10 TIP31 BJT (NPN) Transistor**

**2.11 Potentiometer**

**2.12 Analog and Digital Temperature Sensor**

**2.13 PV (Photo voltaic) Panel**

**3 Hardware Design and implementation**

**3.1 Hardware Interaction**

**3.2 Power Supply**

***Requirement*:***The system shall generate its own regulated supplies from a nominal 9 V-12 V battery or power supply. The system is to generate a 5V and 3.3V supply voltage from the power supply.*



Figure 1: 5v Power Supply Diagram

The system takes as input a 9v nominal supply voltage, from which the 7805 regulator in a T0220 package regulates the input voltage down to 5v. The circuit implementation is depicted in Figure 1 above. Components used in the 5v regulator circuit not only include the 7805-voltage regulator but also capacitors C1, C2, and C3 and diode 1N4007.

The capacitors not only serve to filter out noise and high frequency signals from the power supply ensuring a cleaner output, but also smooth out the voltage, reducing voltage ripples, preventing sudden drops in power supply and providing a more stable DC output.

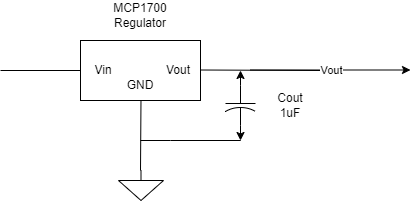


Figure 2: 3.3V Power supply diagram

All the pins on the stm32f411RE micro-controller are not 5v tolerant, for this reason having 3.3V rails is crucial for the rest of the system functionality to be implement. The MCP1700 (3.3v regulator) takes as input the output from the 5v regulator and further reduces the voltage to 3.3V. The minimum and maximum input voltages into the regulator is 2.3V and 6V respectively. The purpose of the capacitor is as described for the 7805-regulator circuit.

**3.3 Push Button**

***Requirement:*** *The system will implement a command interface by using the five push buttons* ***.*** *The push buttons are to be wired as active low, read as inputs by the GPIO pins of the Nucleo Board*

1. *The RIGHT button initiates a calibration procedure*
2. *The LEFT button selects the display mode for the LCD*
3. *The TOP button starts and ends the measurement, calculation, UART transmission, and LCD display of temperatures (Ta, Tsp) and light intensity (L xd) with two presses.*
4. *The BOTTOM button starts and ends the measurement, calculation, UART transmission, and LCD display of PV parameters with two presses.*
5. *The MIDDLE button initiates an RTC clock set menu where TOP and BOTTOM buttons adjust values, and MIDDLE confirms each entry, with the final time reported via UART.*

The buttons are configured as active low configurations, and read as inputs into GPIO pins of the STM32F411RE Nucleo board. On a button press the signal is to be driven low, upon which the system detects that a push button has been pressed. Current limiting resistors have been placed in series with the push buttons, to prevent damage to the Nucleo board.

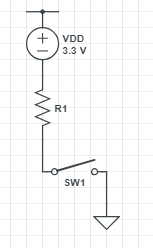


Figure 3: Active Low button configuration

No external circuit need to be built to achieve this active low configuration. The stm32f411 board has a weak internal resistor that pulls the signal high (*STM32F411RE.PDF*). This weak pull-up resistor has a typical resistance of . The top, bottom, left, right and middle buttons are connected to GPIO pins PB9, PB13, PB8, PA7 and PA6 respectively,

**3.4 Debug LEDs**

***Requirement****: The debug LEDs will be used as system state indicators as follows:*

1. *D2 - Power efficiency measurement in progress (flashing – 100ms ON/OFF), and completed (D2 = ON)*
2. *D3 - Temperature and Light measurement in progress (flashing -50ms ON/OFF), and completed (D3 = ON)*
3. *D4 - Calibration sequence in progress (flashing – 200ms ON/OFF), and completed (D2 = ON)*
4. *D5 - Cleanness index progress (flashing – 100ms ON/OFF), and completed (D2 = ON)*

**LED circuit**

The LED circuit serves to indicate to the user the current state of the system. The LEDs are labeled, D2, D3, D4, and D5, connected to pins PB10, PB4, PB5 and PA10 respectively. The LEDS can be in one of two states:

1. Flashing ON and OFF at a specific rate
2. Remain on

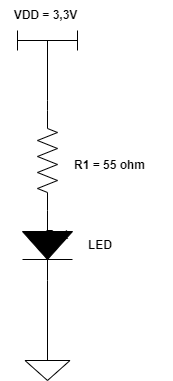


Figure 4: LED circuit schematic

The system flashes the LED when a measurement is in progress. Once a measurement stops, the LED corresponding to the devices that are being measured stop flashing, and remain on, indicating the end of the measurement.

The LED has a forward current (***If***) of and a forward voltage (***Vf***) of . The voltage supplied from the PINS to the LED circuit is 3.3V (*stm32f411re.pdf*). The pin can sink/source a *±8mA*, and can sink/source a maximum of ±25mA (stm32f411re.pdf) . To prevent any damage to the MCU pin the current is restricted to be 20mA. From this an appropriate resistance value is calculate using:

Thus, an appropriate resistance value for the LED circuit of at least 55Ω is required.

**3.5 LCD Display**

***Requirement:*** *The objective of the LCD screen is to present useful information to the system owner. The LCDs must be operated in 4-bit (nibble) mode.*

The system makes use of a 16x2 character LCD display, which is connected to the baseboard using a male header strip soldered to ensure secure attachment. The PCB is designed with pre-connected VCC (5 V) and GND for the LCDs however the LCD used in my designed system (*Micro-robotics LCD1602-WB-33V*) is 3.3 V compatible and powering the LCD from 3.3V results in a high contrast, resulting in the characters being unreadable.

The maximum and minimum rated supply voltage for the LCD used is 4.3V and 3.1V respectively, with a typical operating voltage of 3.3V, To see characters clearer, the backlight it connected. Although connected to a 5v source, the backlight has a forward voltage of 3.3V, and typical forward current of 40mA and no explicit minimum current. Bearing this in mind a current limiting resistor is placed in series with the LED, so as to not damage the LCD. The circuit diagram for the LCD is drawn as:

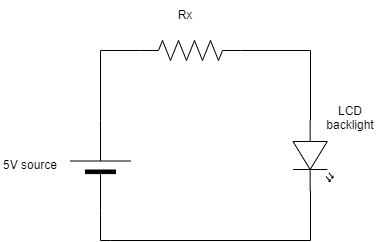


Figure 5: LCD Backlight schematic

The current is limited to 34mA which is the typical currents supplied to the backlight. An appropriate limiting resistor is calculated by:

A current limiting resistor of 50 ohm was calculated. This limits the current to 34mA while at the same time illuminating the lcd at an appropriate intensity for a user to clearly see the values written on the screen.

**3.6 Temperature Sensing circuit (ADC)**

**Requirement:** The system will measure the ambient temperature using the analogue temperature sensor in .

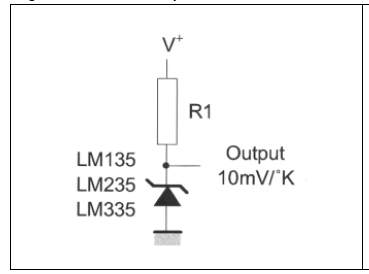


Figure 3.6.1 LM235 Analogue sensor circuit (reference from datasheet)

The LM235 is a precision temperature sensor that can be easily calibrated (reference LM235 datasheet, page 1), and operates within range of current values , having a initial accuracy. The LM235 operates in temperatures of between . The output of the sensor is calibrated at . The analogue sensor circuit is has been designed to be able to measure the ambient temperature over the full range of to .

To be able measure the temperature over the full temperature range, the resistor value R1 is found by relating the temperature over to the expected voltage that would be output by using the provided calibration factor from the datasheet, to relate the temperature to a voltage level.

This step and method to find this resistance is:

1. Find the Kelvin temperature for the lower and upper bound of the given temperature range.

At and the kelvin temperatures are 233.15 and 398.15 Kelvin respectively

1. Using the sensor calibration factor of find the voltages for the lower and upper bound.

For this temperature range the voltages found for the lower and upper bound are 2.3315V and 3.9815V.

1. Using the lower and upper bound voltages found, the valid resistor range is found.

At the minimum resistance, the current is at its maximum of 5mA, and voltage at its maximum. From this, the lower bound of the resistor range is found from the formula:

Similarly at the maximum resistance, the current and consequently the voltage output from the Analogue sensor will beat their minimum. The equation describing this relation is:

The resulting resistor range from which temperature will be measured over the full resolution by the Analogue sensor is . For no particular reason, the resistor chosen for my designed system is .

The output is input into the ADC1 channel 0, which is connected to GPIO PIN PA0, of the NUCLEO BOARD.

**3.7 Lux sensing circuit (Op-Amp)**

***Requirement:*** *The system will measure the light intensity the PV panel is exposed to a photo sensitive diode-based sensor.*

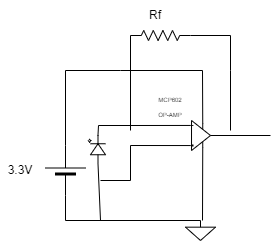


Figure 6: Lux sensing circuit

Measurements will be conducted using an Osram SFH 203 PIN photodiode paired with an operational amplifier to create a photo-current-to-voltage converter circuit. The resulting output voltage will be connected to ADC pin CP4 on the STM32 microcontroller for whether the output is process (see section 4 software implementation).

**3.7.1 MPC602 Op-Amp**

The ADC of the nucleo pin , can take as input a maximum voltage of 3.6V, for this reason, the current to voltage photodiode circuit makes use of MCP602 op-Amp is used to protect the PIN by limiting the voltage output from the photodide to 3.3V.

The operational amplifier (op-amp) in the photodiode current-to-voltage converter circuit is used to convert the current generated by the photodiode into a proportional voltage. When light hits the photodiode, it generates a photo-current, which flows into the inverting input of the op-amp. The op-amp is configured as a transimpedance amplifier , and it uses the feedback resistor Rf to convert the current output from the photodiode into a voltage at the output. The voltage output is given by the equation:

Where is the current generated by the photodiode. The output voltage is fed as input into the ADC input pin of the Nucleo board. The op-Amp ensures high sensitivity and linearity in the conversion, providing accurate measurements of the light intensity.

**3.7.2 Photodiode and feedback resistor calculation**

To determine the feedback resistor, the relationship between the current output from the Photodiode and light intensity is to be determined. This relationship is given by the graph (Add to Appendix):

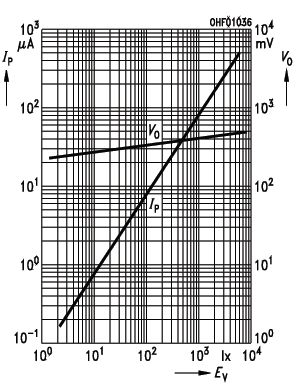


Figure 7: Relationship between Current and Lux (Light intensity)

Bearing in mind the above is a log graph, points (50, 4\*10^-6) and (1000, 80\*10^-6) were used to define the equation of the lux of current. The equation of the straight line of the graph is given by the equation:

Using the points found, the gradient is found to be m= 1. From this we find the constant term (**k**) by making it the subject of the formula resulting in the equation:

The point is substituted above to find . Using the calculated values and designing for 30000 lux, the current expected to be output at this intensity is found to be:

At this maximum intensity, the output voltage should be 3.3V. Using these two quantities, the feedback resistor value is found to be:

This resistor scales the current and lux reading to the appropriate range of [0, 30000] and thus the voltage output from the op-Amp from [0,3.3V]. Note: Because an exact 1375Ω could not be found, a 1200Ω resistor was used.

**3.8 Voltage / Current sensing circuit (PV Panel)**

***Requirement****: The system will use a representative PV panel as a reference PV source. When exposed to light, the PV module generates power that needs to be measured. A circuit will be constructed to measure the output voltage (VPV) and current (IPV) of the PV module. The current will be converted to a voltage measurement so that both voltage and current values can be provided as analog voltage inputs to the STM32 ADC pins.*

To calculate the and determine the current and voltage from the PV panel, a measurement circuit is connected to the terminals of the PV panel. The schematic for the PV panel and connected measurement circuit was provided and is given by the diagram below:

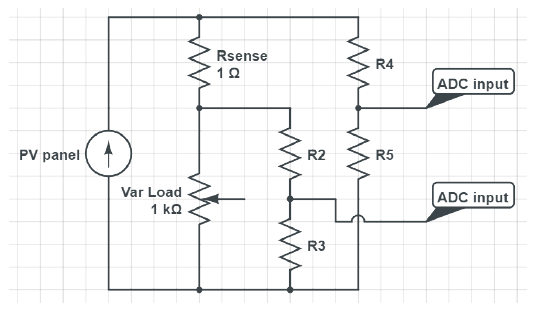
****

Figure 8: PV Panel and Measurement circuit

It is important to note that to be able to measure the voltage and current from the PV panel as accurately as possible, it is required from that the measurement circuit draw as little power as possible from the PV panel. For this reason, a design constraint is imposed, where the measurement circuit is to draw a current of of the current produced from the PV panel.

Note: We’ll denote the terminal at top of the Rsense resistor to be Node A, and the terminal below the Rsense resistor to be **Node B**.

The PV panel produces a current (we’ll denote this as ) at **Node A**, where 1% of the current is to flow from to the circuit to the right of the node (we’ll denote this as current as ), through R4 and R5. Next 99% of the current it to flow through resistor Rsense (we denote this current by to **Node B**. At Node B, the current splits again with current flowing to the right to the measurement circuit (we’ll denote this current by and the remaining current through the **1k potentiometer** (we’ll denote this by ).

Keeping the relationships in mind, the equation to determine resistors R4 and R5 is found to by making the following deductions and conclusions:

1. (when designing, we design for the maximum case)
2. , (which is the voltage between Node A and ground)

Substituting the expression for and in terms of and canceling common terms, the resulting expression is:

1. , (the result of voltage division, where Rx is the resistor to be found)

We note that when the voltage at Node A is at a maximum (8.2V) the voltage at R5 = 3.3V, as the voltage across R5 is a scaled down representation of the voltage at Node A.

Using the above knowledge, and the substituting the values into the voltage divider equation and round to the nearest 10000, R5 is found to be .

1. , (where Rx is the resistor to be found)

R4 being the only unknown, R4 is made the subject of the formula above, and rounding to the nearest 10000 R4 is found to be

Thus, concluding the procedure taken to find R4 and R5. Resistors R2 and R3 are found in the same way.

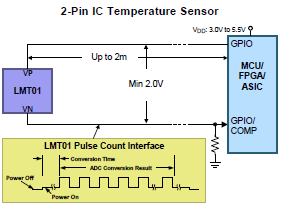
Similar assumptions are made to find Resistors R2 and R3. Where:

Substituting current and and solving the system of equations and rounding to the nearest 10000 results in the values of and .

Thus concluding the methodology used to find the appropriate resistor values for the measurement circuit to as accurately as possible measure the PV panel voltage and current.

**3.9 LMT01 Temperature Sensor**

The LMT01 device is a high-accuracy, 2 pin, temperature sensor with an easy-to-use pulse counts current loop interface. The LMT01 has a pulse count interface which is used to determine the temperature. Where the number of output pulses is proportional to the temperature.

**Figure 9: LMT01 top view and pin Figure 10 LMT01 micro-controller connection**

The LMT01 sensor outputs a current pulse that toggles between a high current of 125µA and a low current of 34µA. The LMT01 takes as input 5V source, with the minimum voltage across the sensor to be 5V, with the output of the sensor (VN) connected to pin PA15 on the MCU. For the MCU pin to be able to detect the output voltage from the pulses, the current is converted to an appropriate voltage by calculating an required resistance value. This value determined by the equation:

The micro-controller detects as input a low signal that is less than **0.3VDD** and an input high voltage (VIH) that is a minimum of **0.7VDD** (*stm32f411re.pdf pg 98*). VDD falls in the range of [1.7V, 3.6V] (*stm32f411re.pdf*). For VDD = 3.6V it is determined that the high input voltage should be greater than 2.52V and the low input voltage should be less than 1.08V maximum. Thus the minimum resistance value for the high current of 125µA is determined by:

For the designed circuit, a resistance value of 22kΩ is chosen which meets the above threshold. To verify that the voltage (**VIL**) from the low current (34µA) is within the maximum threshold of 1.08V, we compute:

To verify the voltage (VIH) from the high voltage is above it’s minimum threshold 2.52V, we compute:

**3.10 Active Load and RC filter**

**4 Software Design and Implementation**

The system is built to handle multiple inputs, each affecting its state. To manage this, regular checks are performed to see which inputs have been received, allowing the system to update its state and respond appropriately.

**4.1 Software Interaction**

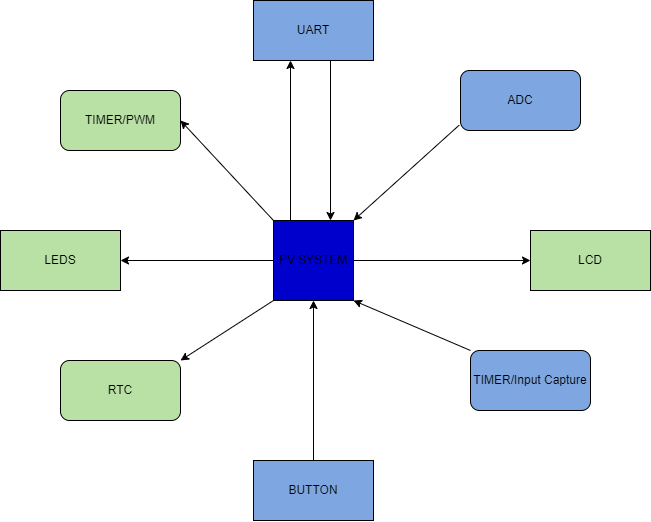


Figure 11: Software block diagram

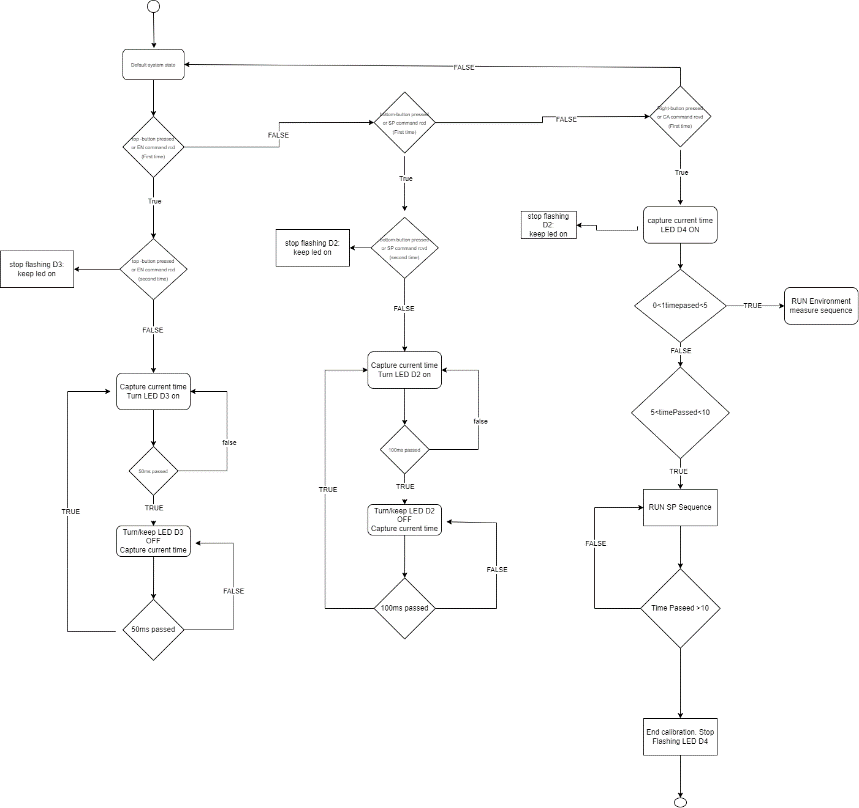
Figure 13, encompasses most of the software interactions that occur in the system. System inputs are denoted by the light blue blocs, system outputs or denoted by the green blocks.

Each of these interactions and more are discussed in the subsections that follow.

**4.2 Debug LED+ system state indication logic**

Debug LEDS, D2, D3, D4 and D5 are used as a means to indicate to the user, the system state at any given time. LED D2 is to flash at rate of 100ms If an environment measure has been initiated, and stop flashing when the when the command has been sent again (either via push-buttons, or UART). Similarly LEDs D3 , D4 and D5, are to flash at the rates specified in section 3.4.

The logic of the state machine responsible for switching between LEDs is in Appendix A.



**4.3 Button bounce handling**

Button debouncing is important because it stops false signals that happen when a button is pressed or released. Without it, a single press could be mistakenly registered as multiple presses, causing the system to behave unpredictably. The state machine that handles debouncing in software is describe in figure 1:

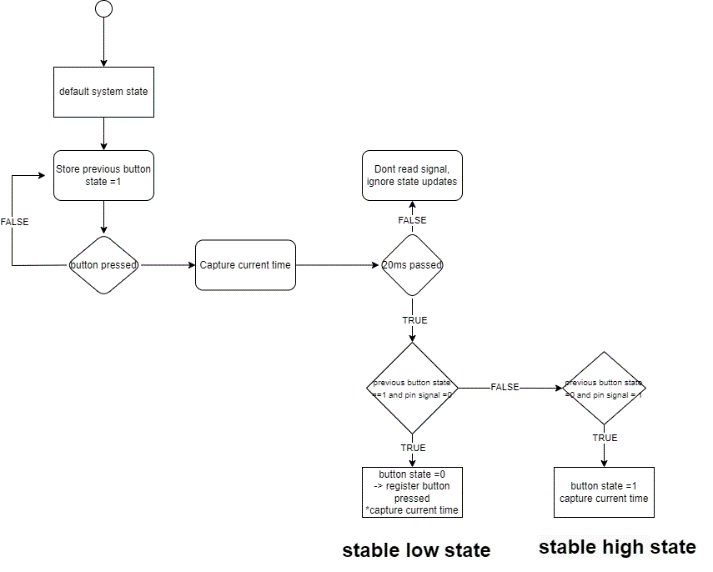


Figure 12: Button debouncing

The left, right, bottom, top and middle button have been connected to pins, PB8, PA7, PB13, PB9,PA6. The pins have been setup as EXTI, to trigger on both rising and falling edge of the signal.

The button circuit is an active low configuration, with the initial state of the signal always high (3.3V). Mimicking this in code, a variable *previous\_button\_state* has been intialised to 1. When a button input has been received, the signal is ignored for 20ms so that the signal stabilizes. After which a check is performed to update the state of the button state variable. This is explained in the button debouncing state machine in Figure 12.

* 1. **UART Communication (protocol, data conversion)**

**Requirement:** *The device shall use a UART connection, in both transmit and receive modes. The device shall operate in 8 data bits, one odd parity bit and one stop bits configuration, at a data*

*rate of 115200 baud. The UART information, both transmitted and received, shall be formatted*

*as per section*

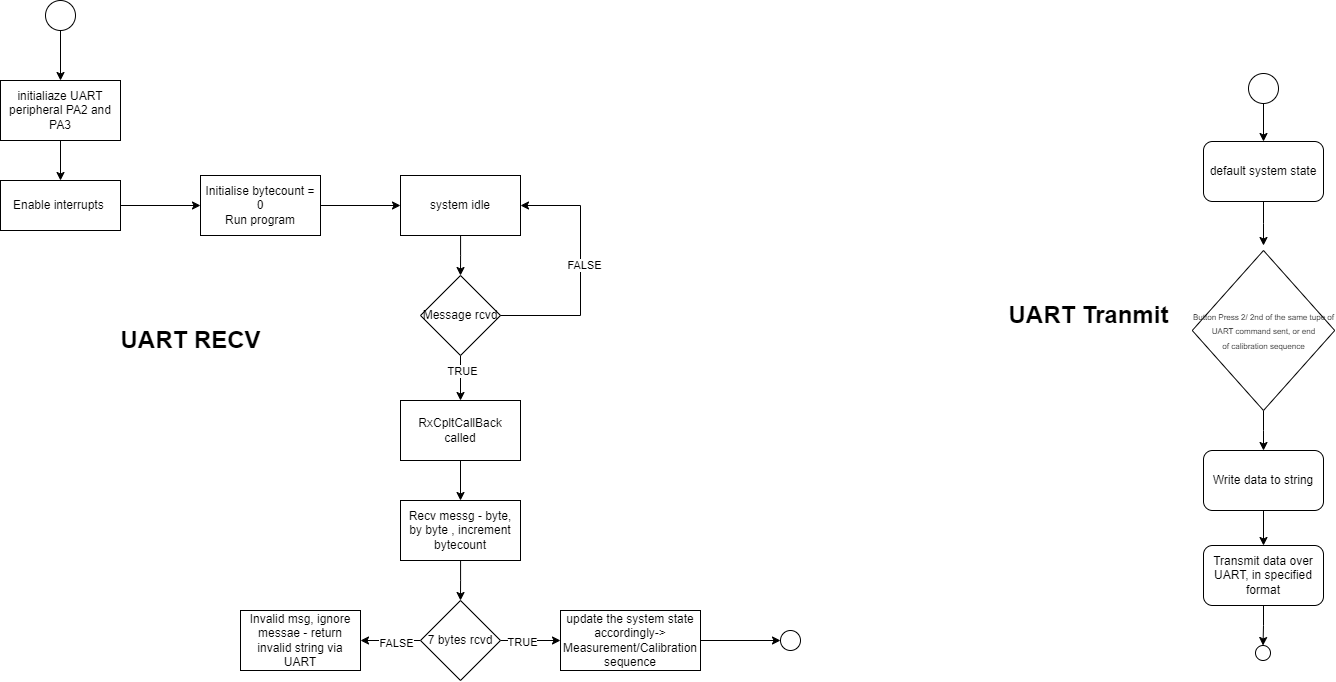


Figure 13: UART Receive and Transmit communication

GPIO PIN PA2 and PA3 have been configured as the UART transmit and receive pins respectively on the STM32Cube IDE. The UART parameter settings have been set to: Buadrate: 115200, Word length: 9 bits (including parity), Parity: odd and stop bits: 1. These setting adhere to the system requirement. The UART Communication has been configured to send and receive messages via interrupts. For this reason the USART2 global interrupt flag has been enable under the NVIC parameter settings.

Using interrupts to check when a message is received, reduces the system burden. Instead of continuously checking if a message has been received in the main loop, we receive messages via interrupts using the HAL\_UART\_Receive\_IT() function which in turn calls the ***HAL\_UART\_RxCpltCallback()*** function that processes received bytes. Within this callback a check is performed to see if a valid command has been received. If true, a flag is set, and the system state is updated based on the command recieved. After receiving a byte, the received bytes get added to an array, after which the receiver is reprimed to receive the next byte. If an invalid valid command has been received, the system ignores the command.

Messages are transmitted via UART using the HAL\_UART\_Transmit\_IT() function. At the end of each measurement sequence, the results of the measurements are stored in a string, after which sent via UART and LCD. This gives the USER a visual representation of the state of the system.

**4.5 ADC, Setup, and channel Management**

In this project, the device measures four single input channels: the PV voltage (Voltage+), the voltage after the current sense resistor to calculate the PV current (Voltage-), the LM235 ambient temperature sensor, and the SFH203 photodiode output.

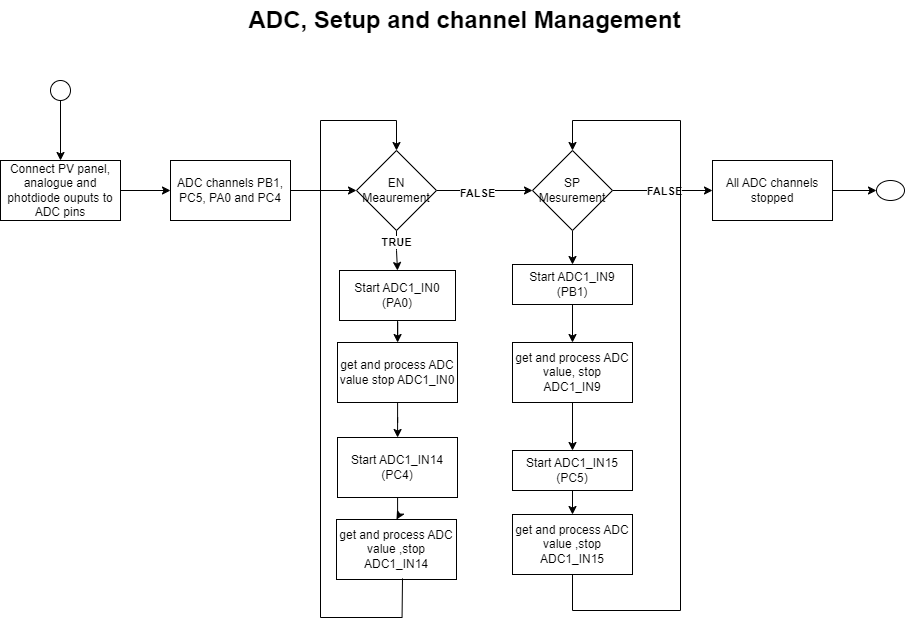


Figure 14: ADC channel Management

GPIO PINs PB1, PC5, PA0 and PC4 have been configured as ADC inputs. Connected to ADC1 channels 9, 15, 0 and 14 respectively. Default ADC parameter values were used for each pin. Because the STM32F411RE Nucleo board has 1 ADC with 16 channels, it was not possible to have each channel running/processing the values from them, since only one adc could be active at any given time.

The solution to managing these different channels was to allow only one channel to be active at any given time. Depending on the state of the system, i.e. environment measure or panel measurement, to ADC channels would have to be read, and have their values processed. During the environment measure operation, channel 0 and 14, connected to the outputs of lm235 and sfh203 photodiode are to be read. During the Panel measurement operation ADC channels 9 and 15 are connected to the Voltage+ from the PV panel and the Voltage- from the Rsense resistor.

The next section discusses the processing these values read from the adc to determine the ambient temperature, current, lux and voltage.

**4.6 Processing (convert & calibrate): Temp, lux, V,I,P, Active Load, Efficiency**

**4.7 LCD Interface (control, data conversion)**

**4.8 RTC handling and setting procedure**

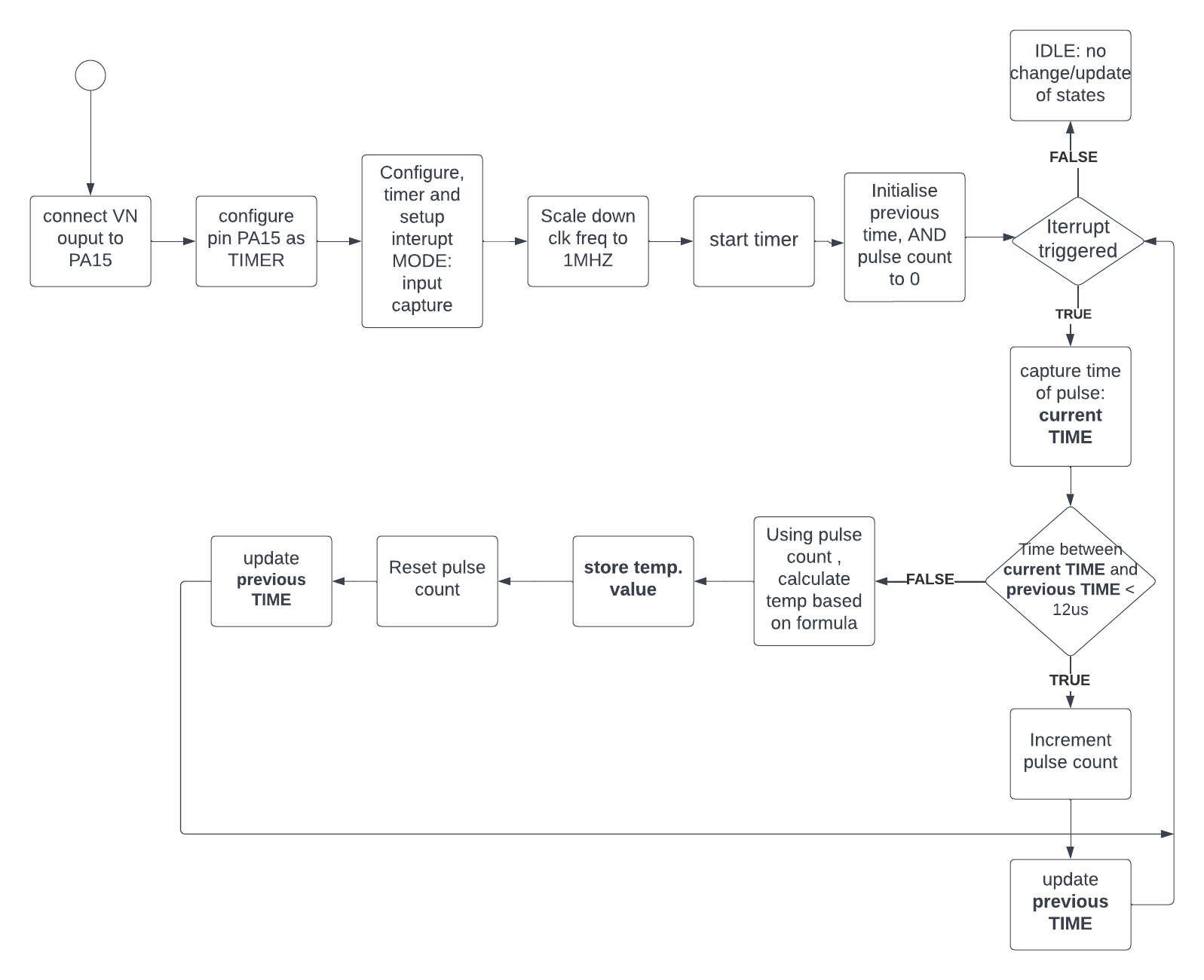
**4.6 LMT01 Interface, timing, synchronisation**

**LMT01 description**

The LMT01 had a window period of maximum 104ms in which the measured temperature by the device will be output as pulses. At every 104ms window period, the temperature is to be determined, while being cautious of overlapping window period readings.

The output of the average temp of the LMT01is to be within 3 degrees of the measured temperature of the testing station. The temperature from the LMT01 sensor is to change appropriately when the sensor is touched. This requirement is met and is proved in the next section.

**Software design**



**Figure 5: LT01 software design – Measurement and Conversion**

To setup the software for the LMT01 sensor we begin by configuring the required peripheral for the LMT01 sensor via the STM32CUBEIDE ***.ioc*** file. GPIO Pin PA15 is set as timer PIN, with interrupts enabled by checking the NVIC box in the parameter settings. Next this interrupt is configured to be triggered on the rising edge with the mode of the timer to be in Input Capture mode for reasons that follow.

TIMER PINS can be set in one of 4 modes, namely; Input Capture, PWM mode, One Pulse Mode and Output compare Mode, each with their own functions. Of interest is the Input capture. This mode is to be used to capture the time at which interrupts occur, which enables us to keep track of the windowing periods. PIN PA15 is setup as a timer in input capture mode, where PA15 is connected to timer 2 CHANNEL 1.

The hardware has been setup such that the output current from the LTM01 is converted to valid voltage logic levels. Each pulse will trigger the interrupt. The LMT01 sensor outputs pulses at 88khz, from this the calculated period of each pulse is approximately 11.36us ~ 12us. Because the period of each pulse is in the micro-second range, the timer clock is configured to count in **µs** (micro-seconds). For this, Timer 2 (timer that PA15 is connected to) frequency is scaled down from 84Mhz to a 1Mhz signal.

The timer is started and the approach taken is to measure the time between consecutive pulses. Noting that within a window period, a pulse is expected every 12µs once the pulse train begins. For any measured time larger than 12µs, it means that the next pulse window has started, thus the count is to be restarted and the temperature to be recalculated.

Within each window period, that temperature of that window is determined and only at the start of the next window period.

After the temperature has been converted, it’s value is stored (ready to be processed), the pulse count is reset, and the pulses at the next window period begins incrementing with every pulse received and again at the start of the next window period, the temperature is again re-calculated for previous window period.

The temperatures are calculated based off the formula, where PC is the number of pulses:



**Figure 6: Temperature calculation formula**

This process is repeated indefinitely. Note that all of this is happening inside the interrupt handler function. Where the interrupt is triggers at every pulse recored. The stored temperature is ready to be used for further processing at any time.

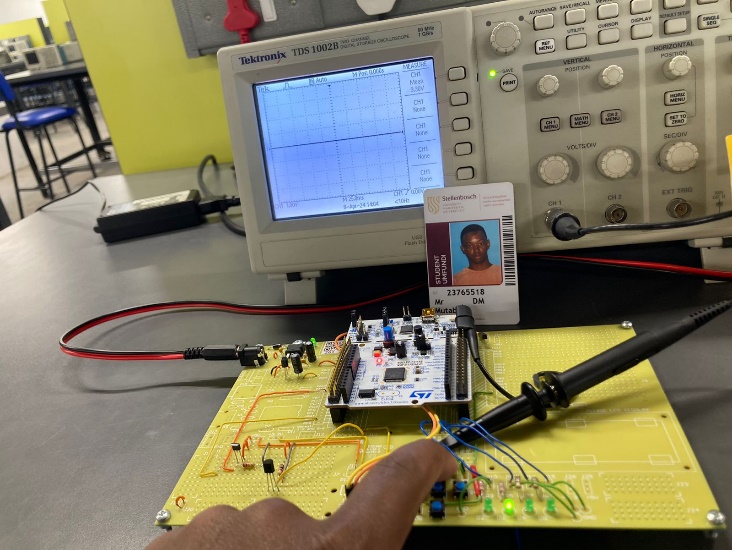
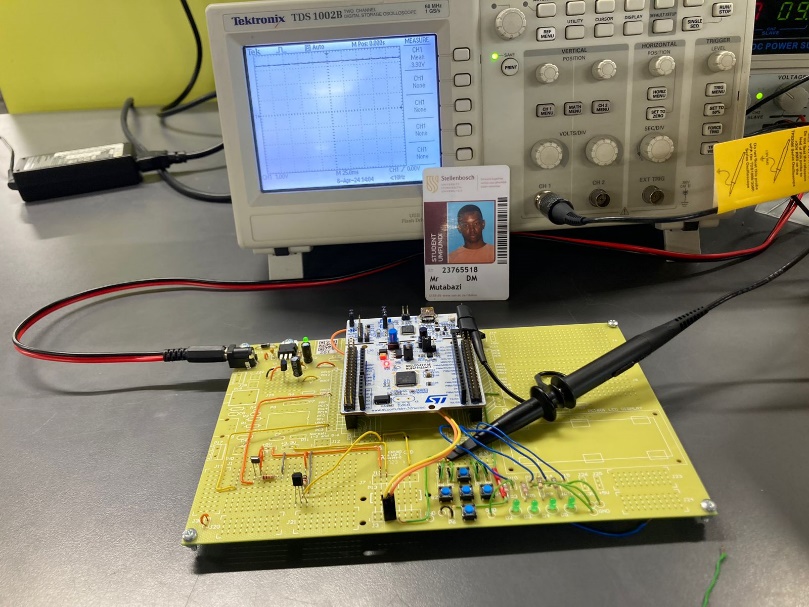
****

**Figure 7: Input capture interrupt callback function**

**5 Measurements and Results**

* 1. **Power Supply**
  2. **UART Communication**
  3. **Button Circuit**

**Top push button - active low**

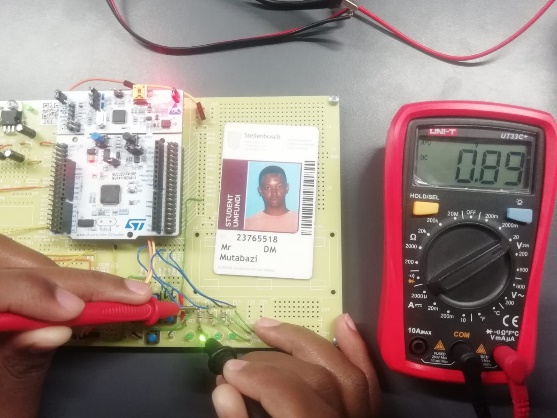


**Figure 11: Button Not pressed Figure 12: Button Pressed**

Measuring using a multimeter/and oscilloscope, Figure 11 shows us that when the top button is not pressed the pin is at a high signal. This confirms the presence of an internal pull-up resistor pulling the signal high. When the button is pressed, Figure 12 shows us the state of the signal, dropping low. From this we can confirm the active low circuit connection.

* 1. **LED circuit**

**LED circuit**

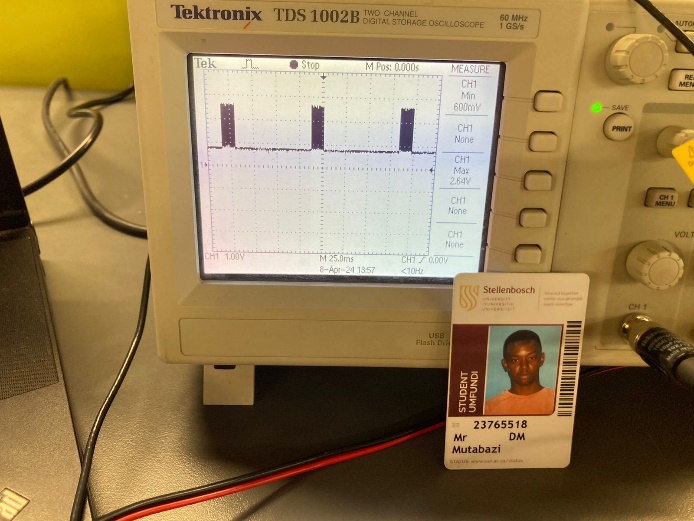
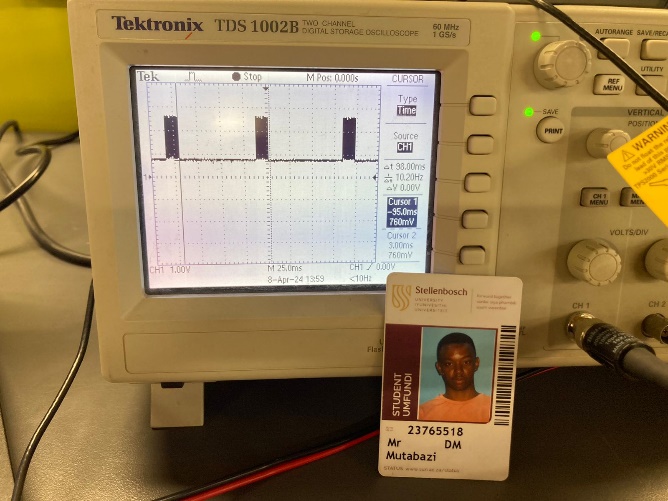


**Figure 12: Voltage across LED**

Figure 12 illustrates the voltage measurement across the resistor to be 0.89V, from which the current is measured using ohms law. The resulting current in the circuit is **15mA** which is less than maximum current supplied by the pins of 25mA

* 1. **ADC**
  2. **LCD**
  3. **PV Panel**
  4. **Photodiode and Op-Amp**
  5. **LMT01**

The above image are the pulse outputs for the LMT01 sensor, where the pulse window period is roughly almost 104ms as shown by the oscilloscope. Note that it is not exactly 104ms as it is not clear after the pulse go low, when the next period starts. It is also worth noting that indeed the pulse period is roughly 88khz.

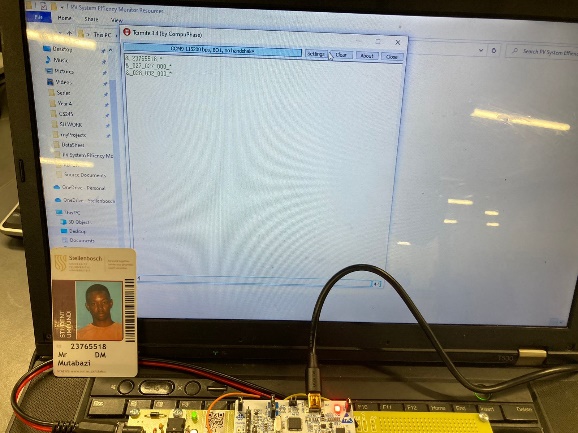
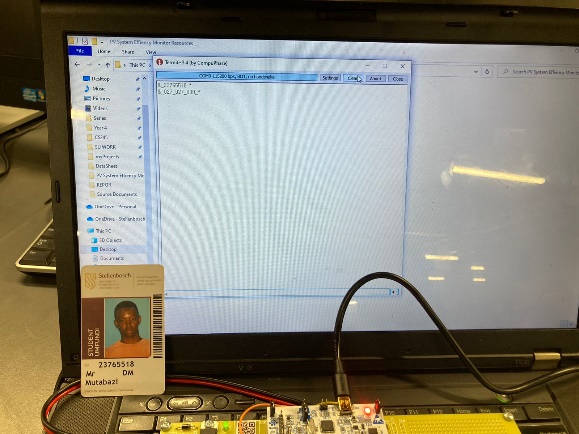


**Figure 8: LMT01 pulse outputs and low and high voltage**

The measured high voltage and low voltage was measured with an oscilloscope, showing that the voltages were at the appropriate voltage levels to be detected by the STM32 pin. Where the high voltage was **2.75v** and low voltage **0.75V**. Figure 8 above shows this:

The above proves that for the low and high output current from the LMT01, using a resistor value of 22kΩ they have been converted to valid logic level, able to be detected by the MCU.

* 1. **LMT01 Functionality – Reaction to changing temperature**



**Figure 13: LMT01 sensor: Not touched Figure 14: LMT01 sensor: Touched**

From the termite output, we observe the temperature measured by the LMT01 when it is not touched (Figure 13) and when it is touched (Figure 14). From this we observe that the sensor temperature changes appropriately when changed and meets the necessary requirements.

Based on the results produced from these measurements, the system meets the necessary requirements.

* 1. **Complete System**

**6 Conclusions**

**6.1 Non-Compliance and Short comings**

**6.2 Possible Improvements**

**7 Appendix A**

**REFERENCES**

[1] STMicroelectronics, "RM0383, Reference manual: STM32F411xC/E advanced Arm®-based 32-bit MCU," [Online]. Available: [https://www.st.com].

[2] STMicroelectronics, "STM32F411xC STM32F411xE Arm® Cortex®-M4 32b MCU+FPU, 125 DMIPS, 512KB Flash, 128KB RAM, USB OTG FS, 11 TIMs, 1 ADC, 13 comm. interfaces," [Online]. Available: [https://www.st.com]