DESIGN (E) 314
PRELIMINARY REPORT

PV System Efficiency Monitor

Author:
Dgibrilly M. Mutabazi

Student Number:
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.
2. 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.
3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.
4. 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.
5. 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.

D.Mutabazi	23765518
Handtekening / *Signature*	Studentenommer / *Student number*
D.M Mutabazi	01/04/2024
Voorletters en van / *Initials and surname*	Datum / *Date*

Table of contents

Task 1: Hardware Design Details

LMT01 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.

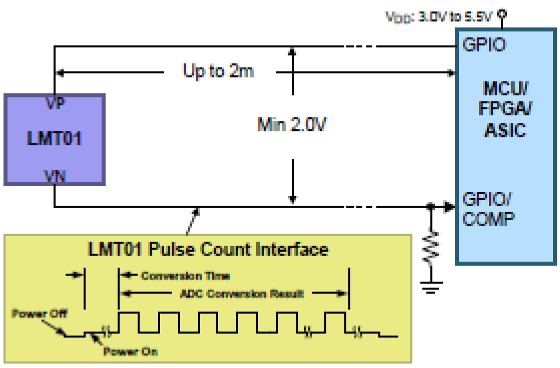

Figure 1: LMT01 top view and pin **Figure 2: LMT01 micro-controller connection**

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:

$$V = IR$$

The micro-controller detects as input a low signal that is less than **0.3VDD** and an input high voltage (V_{IH}) 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:

$$R \geq \frac{2.52}{125 * 10^{-6}}$$

$$\boldsymbol{R} \geq 20160\Omega$$

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

$$\begin{aligned} V_{IL} &= I * R \\ &= (34x10^{-6})(22x10^{3}) \\ &= \mathbf{0.748\ V} \end{aligned}$$

To verify the voltage (V_{IH}) from the high voltage is above it's minimum threshold 2.52V, we compute:

$$\begin{aligned} V_{IH} &= I * R \\ &= (125x10^{-6})(22x10^{3}) \\ &= \mathbf{2.75V} \end{aligned}$$

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.

Top push button

The top button is used to initiate a measurement command across the LMT01 sensor discussed above and the LM235 analog sensor (not to be discussed). Upon pressing the top button once, it is to begin the measurement sequence, after which when the button is pressed again, the system is to stop measuring and returns the measured temperature value from the LTM01 sensor.

The button is configured in an active low configuration. Where on a button press the signal is to be driven low, upon which the system detects the top push button has been pressed.

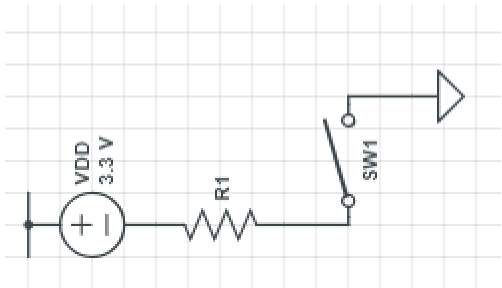

Figure 3: Active Low button configuration

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 $40k\Omega$. The top button has been connected physically to GPIO PIN PB9, which has a supply voltage of 3.3V.

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

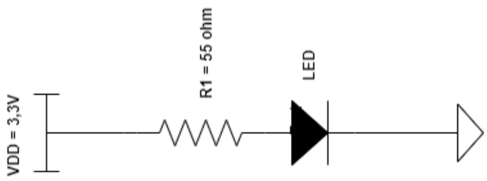

Figure 4: LED circuit schematic

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

The LED has a forward current ($\boldsymbol{I_f}$) of $20mA$ and a forward voltage ($\boldsymbol{V_f}$) of $2.2V$. 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:

$$V = IR$$
$$(3.3 - 2.2) = (20x10^{-3})R$$
$$\boldsymbol{R = 55\Omega}$$

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

Task 2: Software Design Details

LMT01 desription

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

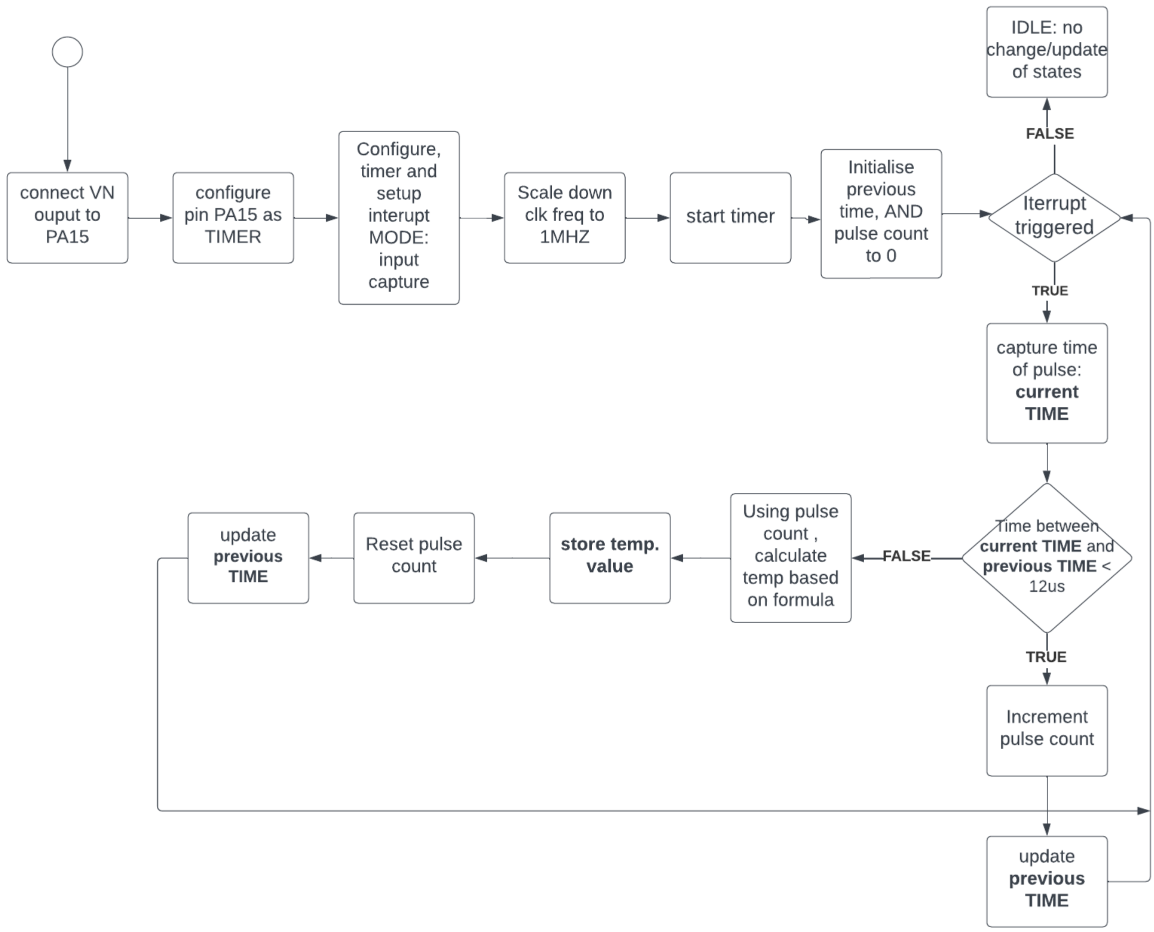

Figure 5: LT01 software design – Measurement and Conversion

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:

$$\text{Temp} = \left(\frac{\text{PC}}{4096} \times 256°\text{C} \right) - 50°\text{C}$$

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.

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
```

Figure 7: Input capture interrupt callback function

Task 3: Testing of system to verify performance/functionality

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.

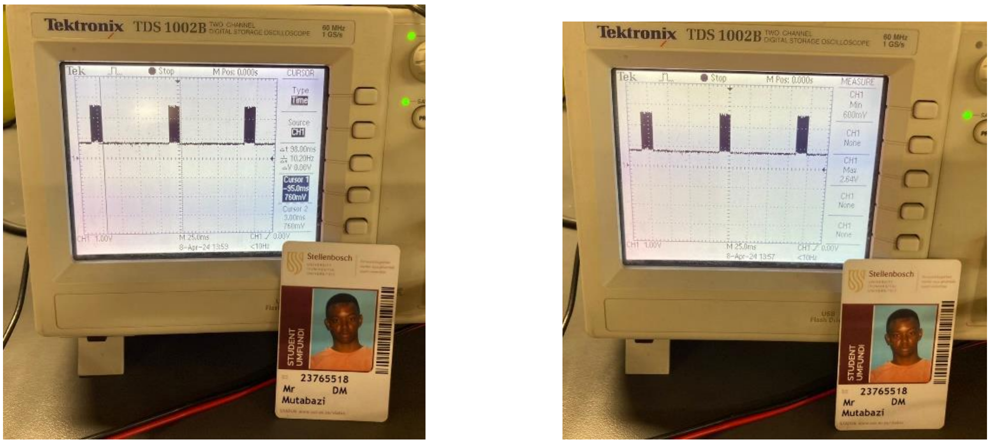

Figure 8: LMT01 pulse outputs and low and high voltage

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 9 below shows this:

Top push button - active low

Figure 11: Button Not pressed

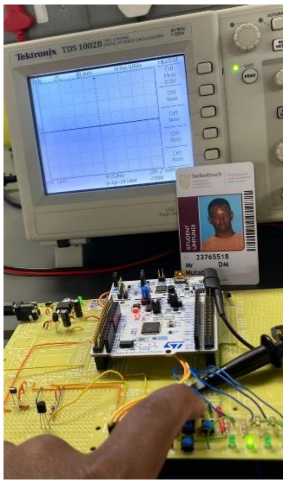

Figure 12: Button Pressed

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.

LED circuit

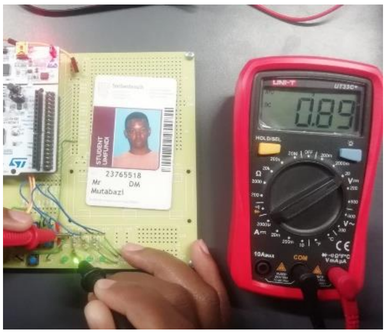

Figure 12: Voltage across LED

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

LMT01 Functionality: reaction to changing temperature

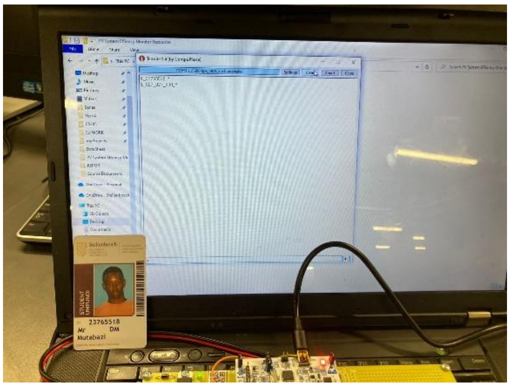

Figure 13: LMT01 sensor: Not touched

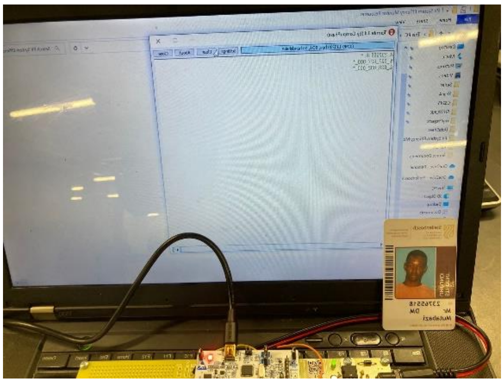

Figure 14: LMT01 sensor: Touched

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.

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]