Milestone 2 Application Note

Richard Dell and Nick Riggins
Rowan University

November 26, 2019

1 Design Overview

For this project, the goal was to write code for a MSP430G2553 and create a circuit that would be able to read the temperature of a voltage regulator using a proportional to absolute temperature sensor, and control a fan to cool down the voltage regulator to the desired temperature. Code was written to enable UART on the msp board to be able to send and receive information about the temperature to the user. The software realterm was used to send a desired temperature to the MSP430G2553 and also to receive the temperature once it was within 2°C of the desired temperature. In order to read the temperature from the PTAT, an analogue to digital converter was used to change the voltage signal from the PTAT to a digital value. This digital value was then converted into a temperature in Celsius based on a series of tests that was done to ensure the correct values. To control the fan, different PWM signals were written in code to be able to change the speed of the fans rotation, and thus increase or decrease its cooling on the voltage regulator. To determine the speed of the fan needed to reach and maintain the desired temperature, a series of tests were conducted and a function was written in C using the results to change the fan's duty cycle to the necessary value.

1.1 Design Features

These are the design features:

- Utilize Hardware PWM
- UART Communication
- Fan Speed Control
- Heat Sink for Heat dissipation

1.2 Featured Applications

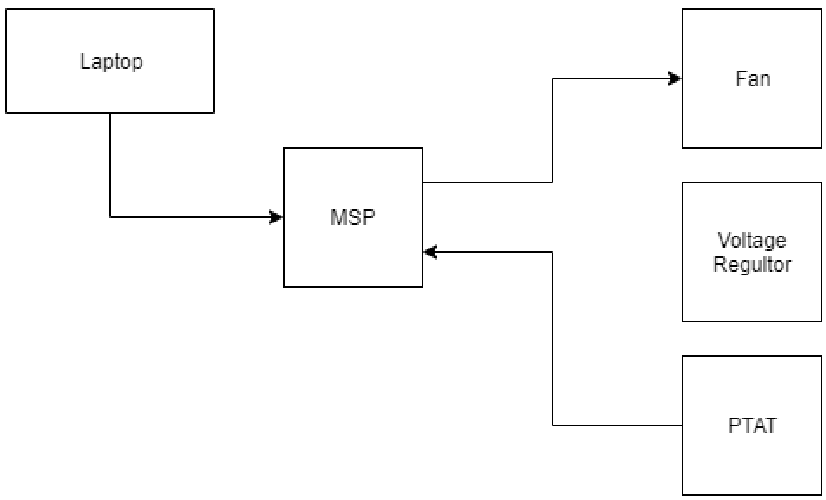
These our the featured applications

- Measure Temperature using a PTAT
- Temperature Control
- ADC conversions

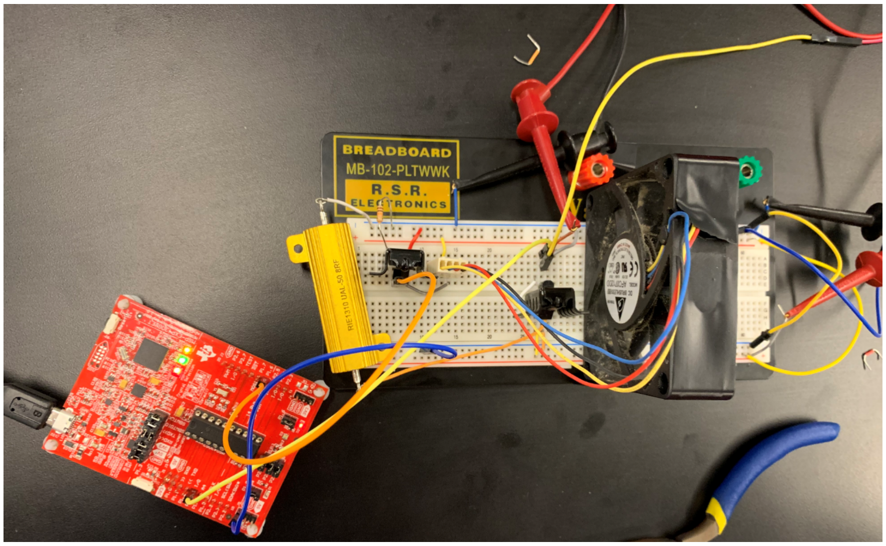
1.3 Design Resources

Github repository link:
https://github.com/Intro-To-Embedded-Systems-RU09342/milestone-2-elite

1.4 Block Diagram



1.5 Board Image



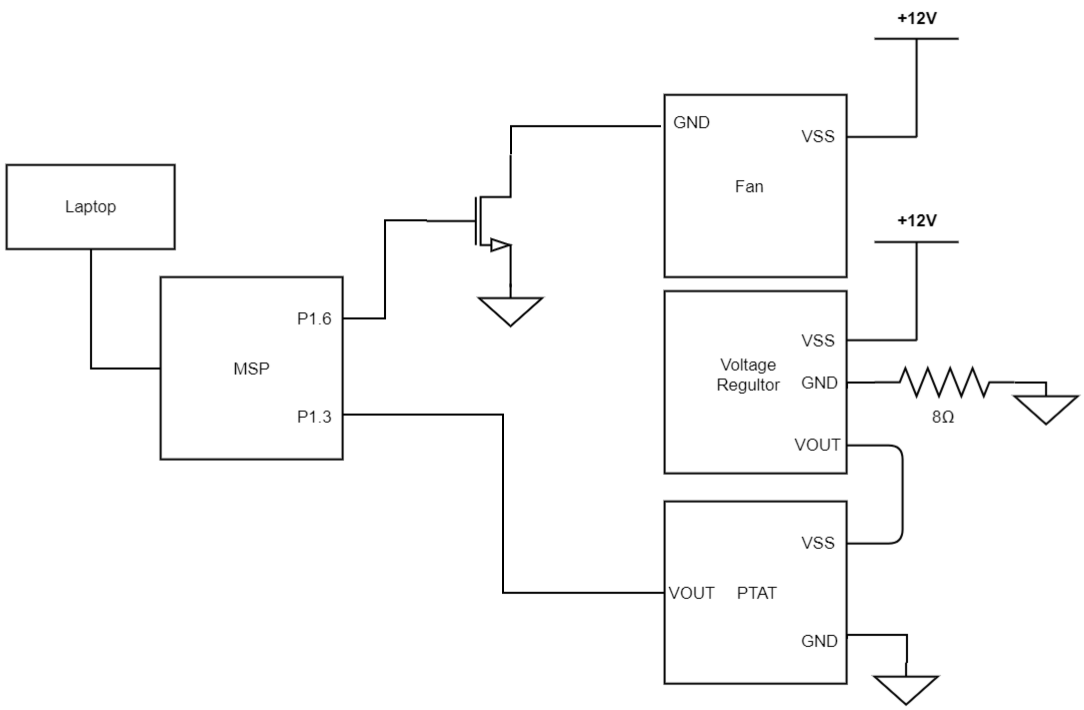
2 Key System Specifications

Parameter	Specification	Details
Fan	Be able to control the speed of the fan	Utilize hardware PWM to change the speed of the fan based on the desired temperature received from the transmission
UART Transmission	Send and receive UART transmissions between the MSP430G2553 and a PC	Receive a temperature that the user wants to set the regulator to, and output the current temperature before the fan starts cooling, and after the regulator has reached the desired temperature
PTAT Reading	Read a voltage value from a PTAT and convert it into an accurate temperature in celsius	Receive the voltage value from the PTAT and convert this value into a digital value using and ADC, then convert this value into an accurate temperature based on a conversion factor received through testing
Temperature Control	Change the Temperature of the voltage regulator to a given desired value and maintain that temperature	Compare the current temperature of the voltage regulator to the desired temperature and change the speed of the fan to maintain the desired temperature

3 System Description

The problem for this project is to be able to cool a voltage regulator that is overheating. The system must be able to control a fan at different speeds to cool the regulator and hold its temperature at the desired temperature that the user inputted into the system. In addition to cooling the fan, the system must also be able to read and output the current temperature of the regulator so that the user will know if the system is cooling properly.

3.1 Detailed Block Diagram



3.2 Highlighted Devices

- MSP430G2553

  This is the micro controller that was used to execute the code from controlling the fan and also sending and receiving the temperature values between the system and the user.

- AFC0712DD (Fan)

  This is the 12 volt DC brushless fan that is used to regulate the voltage regulator's temperature.

- TMP36GZ (PTAT)

  This is used to be able to read the temperature of the voltage regulator.

- IRL520 (NMOS)

  This is used to allow the micro controller to control the pwm of the fan without exposing it to the 12 volts that is driving the fan.

- LM340T57805(5v Voltage Regulator)

  This is a 5 v voltage regulator and it was giving a voltage of 12 v which caused the voltage regulator to over heat.

3.3 Device/IC 1

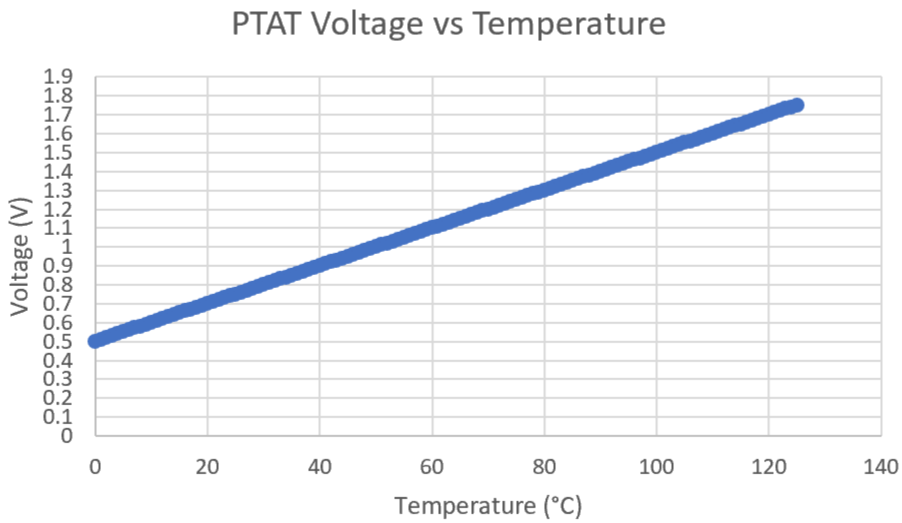
The MSP430G2553 was the micro controller that would be executing the code from the system. All of the code for the system was written in C. The RXD and TXD jumpers on the board had to be switch from connecting vertically to connecting horizontally in order to enable the hardware pwm function of the micro controller. Realterm was used to send and receive temperatures to the system to control it (See ”Getting Started Software/Firmware” section.). The code written for the msp to execute would allow it to read a voltage from the PTAT and convert it into a digital value, then convert it again into an actual temperature in Celsius(See ”temperature Conversion/Control Testing” section). In addition to reading the temperature, the code would also allow the user to input a desired temperature to the system. The msp would then send the current temperature of the regulator to the user before comparing the desired temperature to the actual temperature of the regulator received from the PTAT, and adjust the pwm of the fan according to a set range of temperatures that were found during testing (See ”Temperature Conversion/Control Testing” section). The pwm signal was created by setting up a timer on the msp and having it trigger an interrupt at different times according to the duty cycle. The maximum value for a 100% duty cycle was set to 255, and the other duty cycle interrupt values were derived from this maximum value. After changing the pwm of the fan, the system would then wait for the regulator to get within 2°C of the desired temperature before sending the current temperature of the regulator to the user.

3.4 Device/IC 2

The AFC0712DD Fan was used in order to control the temperature of the voltage regulator. This is a 4 input fan however, for the purpose of this project only the VSS and ground inputs were used. The VSS pin of the fan would always be receiving 12 volts, while the ground pin would be controlled by a mosfet that was driven by the micro controller. It was connected in this fashion to be able to control the speed of the fan with the pwm output of the micro controller. In order to run the fan at 100% duty cycle as intended, the fan would have to be supplied with 0.75 amps of current. At 100% duty cycle, the fan also creates enough airflow to push itself, meaning that the fan itself would have to be secured to be able to use it effectively.

3.5 Device/IC 3

The TMP36GZ PTAT was a 5 volt temperature sensor that was used to be able to read the current temperature of the voltage regulator. The PTAT is able to create a linear relationship between the voltage that it is outputting and the temperature in Celsius. The particular PTAT that was used for this project was rated to output 10 mv/°C and have an offset of 0.5 V, meaning that for 0°C the PTAT would output 0.5 V and each degree that the temperature rose, the output would increase by 10 mV. The outputted voltage from the PTAT would be sent to the msp to be converted from analogue to digital, and then to the temperature in Celsius for the system to use.



3.6 Device/IC 4

The IRL520 is an N-channel power mosfet device that was used to enable the micro controller to control the speed of the fan without exposing it to the higher voltage driving the fan. This mosfet is rated to have maximum value of 16 V DC and 9 amps of current. This particular mosfet was chosen because of its ability to function at higher current values. In order for the fan to work properly at 100% duty cycle, it would have to be supplied with at least 0.75 amps of current, which this mosfet would be able to handle.

3.7 Device/ IC 5

The LM340T57805 is the 5 V voltage regulator that was used to be overheated and then cooled down. This voltage regulator is rated for 5 V, so to get it to overheat a higher voltage was applied to the device. In this case 12 V was applied to the device. The PTAT was used to measure the temperature of the voltage regulator and then a fan was used to cool it down to the desired temperature. Due to the size of the fan, to effectively cool down the voltage regulator a heat sink is added on to the device so it can efficiently dissipate the heat.

4 SYSTEM DESIGN THEORY

The design for this project can be split into four main requirements, reading the temperature of the voltage regulator, controlling the speed of the fan, communicating with the user, and controlling the temperature of the regulator. In order to perform its desired function, the micro controller must receive a temperature from the user for the desired temperature of the voltage regulator. The micro controller would then read the temperature of the voltage regulator and compare the current temperature to the desired temperature. Based on the comparison result, the micro controller would change the speed of the fan to be able to regulate the temperature of the voltage regulator to the user's desired value. Lastly, the micro controller would continue to read the current temperature of the voltage regulator unit it is within 2°C of the desired temperature, and then send the user the new temperature. After changing the temperature of the voltage regulator once, the micro controller will hold the voltage regulator at the desired temperature until a new temperature is inputted again.

4.1 Design Requirement 1

The first design requirement is to be able to accurately read the current temperature of the voltage regulator. This was done using a PTAT that would linearly relate the voltage that it outputted to the temperature of the regulator in Celsius. The PTAT would be set up next to the voltage regulate so that it would be toughing the regulator at all times and get an accurate reading of its temperature. The output of the PTAT would then be connected to pin 1.3 on the micro controller to send the voltage to be read. The code that was written for the micro controller would then act as an ADC and convert the voltage that the PTAT outputted into a digital number. The conversion for this can be set up as a proportion below. Once the value was converted into a digital number, it would then also have to be converted into an actual temperature value in Celsius. In order to convert it into Celsius the offset of the device would have to be taken into account a conversion factor would have to be created so that the digital value could accurately portrait a temperature. The offset was given by the PTAT datasheet and converted to a digital number and then tests were run to determine what the conversion factor would be (Test results can be found in the ”Temperature Conversion/Control Teting” section.). Once the conversion factor was found, the output voltage of the PTAT could be read continuously and converted into a temperature in Celsius for the user to see.

$$\frac{3.3V}{1024bits} = \frac{PTATVoltage}{Nbits}$$

4.2 Design Requirement 2

The second requirement of the project is to be able to use the micro controller to control the fan at varying speeds. This was set up using the hardware pwm function of the msp430g2553. In order to enable this, two sets of jumpers connecting the TXD and RXD pins on the msp would have to be switched from connected vertically, to connecting horizontally. Once the jumpers were switched code could be written to change the dty cycle of the fan. First, a timer would be set up to count to a maximum value of 255 and trigger an interrupt, and the output mode of the timer would be changed to a set and reset mode that would enable the pwm. Next, another interrupt would be created for the timer that would vary based on the user's desired temperature. This interrupt would control the duty cycle of the fan and change it based on the temperature that the voltage regulator needs to get to. After the interrupts were created, an output pin had to be set up to send the pwm signal to the fan, pin 1.6 was used for this. With the pwm output set a circuit must be created so that the micro controller would not be exposed to the 12 volts that was powering the fan. To isolate the micro controller from the 12 volts, a n-channel mosfet was used. By connecting the fan's ground to the drain pin of the mosfet and ground to the source pin, the micro controller would be able to successfully change the speed of the fan when connected to the gate terminal of the mosfet.

4.3 Design Requirement 3

The third requirement of the project is to be able to communicate with the user so that they could input a desired temperature for the voltage regulator and can also see the current temperature of the regulator as well. In order to transmit and receiver information, the micro controller was set up to use UART and realterm was to be used to see the transmissions. Code was written to initialize UART and change the baud rate to 9600. Next, an interrupt would be created to occur whenever the user inputted a temperature. This interrupt would first change the desired temperature of the voltage regulator and then output the current temperature to the user. After the current temperature is sent, it would then run the temperature control function to change the temperature to the desired value and output the current temperature again when it is within 2°C of the desired temperature.

4.4 Design Requirement 4

The final design requirement is to be able to control the temperature of the voltage regulator using all of the previous requirements. The user would be able to input a temperature that they want the regulator to operate at, and the system would have to be able to change the temperature to that and maintain it. After setting up the previous requirements, code had to be written so that the micro controller would know what duty cycle the fan would have to operate at to maintain the desired temperature. To find these duty cycle values, several test were run to determine what temperature different duty cycles would maintain (See "Temperature Conversion/Control Testing" section for more information.). Once these tests were finished, several if else statements were written in C to tell the micro controller to change the duty cycle of the fan based on what temperature range the desired temperature was in.

Duty Cycle (%)	Temperature Range (°C)
100	T >= 27
90	27 <T <= 28
80	28 <T <= 30
70	30 <T <= 31
60	unused
50	31 <T <34
40	34 <= T <36
30	36 <= T <38
25	38 <= T <41
20	41 <= T <45
15	45 <= T <49
10	T >49

4.5 Temperature Conversion/Control Testing

Duty Cycle (%)	Voltage (V)	Temperature (°C)	ADC Value	ADC Value - Offset
100	0.77	27	238	83
90	0.78	28	242	87
80	0.80	30	248	93
70	0.81	31	251	96
60	0.82	32	254	99
50	0.82	32	255	99
40	0.84	34	260	105
30	0.86	36	267	112
25	0.88	38	274	119
20	0.91	41	282	127
15	0.95	45	296	141
10	0.99	49	307	152

The table above contains the results from a series of tests pertaining to what temperature the fan could hold the voltage regulator at with varying duty cycles. The offset ADC value was found after converting the 0.5 V offset of the PTAT into a digital number that our system could use, and was found to be 155. Once the offset was found, a final conversion equation could be made to change the ADC value into an accurate temperature in Celsius using algebra.

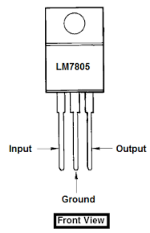
$$\frac{ADCValue - 155}{3.1}$$

5 Test Setup

To test this project you first must setup the circuit. The following steps are to be taken in order to properly recreate the system.

- Connect the voltage regulator

  The input pin must be connected to a 12 V power source, the ground pin to ground. The output pin can be left for now.

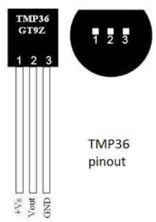


- Connect the power resistor

  The resistor must be set up to connect the output of the 5 V regulator to ground.

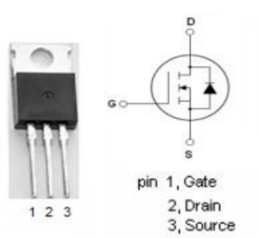
- Connect the PTAT

  It is important to ensure that the PTAT has contact with the voltage regulator to provide accurate readings. The +Vs pin must be connected to the 5 V ouput of the voltage regulator. Then the Ground pin must be connected to ground, and the Vout pin must be connected to pin 1.3 on the micro controller.



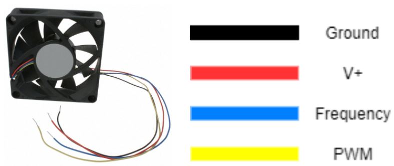
- Connect mosfet

  The gate terminal of the mosfet must be connected to pin 1.6 on the micro controller. The Source pin must be connected to ground, and the Drain pin can be left unconnected for this step.



- Connect the fan

  It is also important for the fan to be placed close to the regulator and secured down to ensure it works as intended. The fan's ground wire must be connected to the Drain terminal of the mosfet, and the V+ wire must be connected to the 12 V power source. The frequency and PWM wires for the fan are left unused for this system and thus, do not need to be connected to anything.

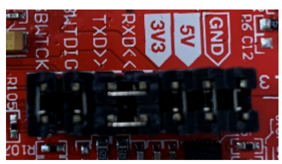


- Connect all grounds

  Connect a ground pin on the micro controller to the ground of the board. Ensure that anything that has a connection to ground is connected to the same ground line (micro controller, voltage regulator, mosfet, PTAT, fan).

- Enable hardware pwm

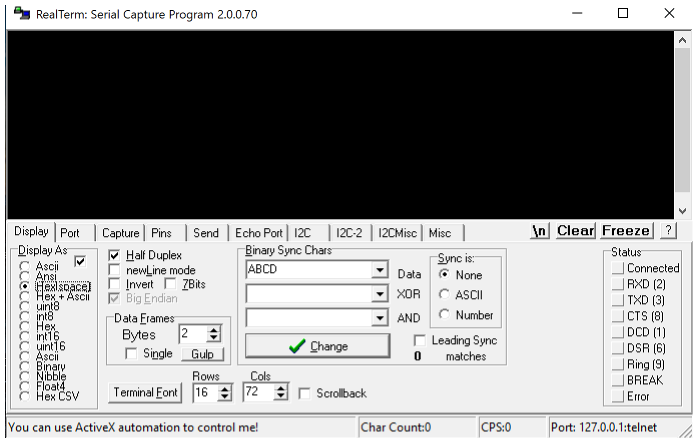
  If the jumpers connected the RXD pins and the TXD pins on the MSP430G2553 have not been changed, they must be changed to connect the pins horizontally rather than vertically.

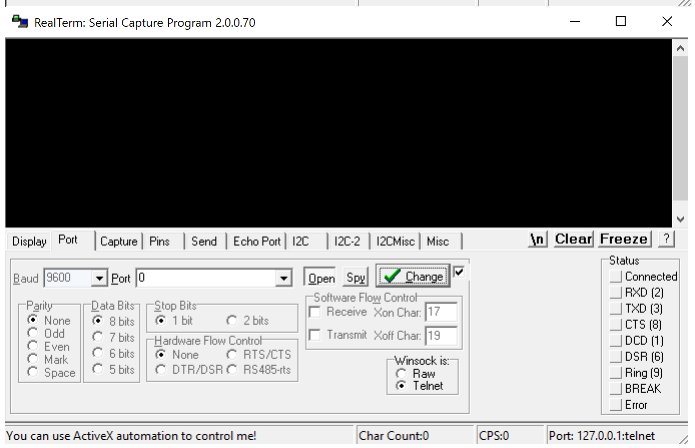
  

After the circuit is setup the micro controller can be connected to the computer though the usb port. Then the program can be ran and then using realterm the temperature can be sent to the MSP430G2553.(See section 5.1 and 5.2 on how to set up and use realterm and the how to use the device)

5.1 Getting Started Software/Firmware

The code for the MSP430G2553 was written in C using Code Composer Studio. After the code was loaded onto the micro controller, Realterm is then used to send the desired temperature and receive the current temperature of the voltage regulator. In order to communicate successfully with the device and send it a transmission the software Realterm is used. Several things must be changed in Realterm for the software to communicate with the micro controller. First in the display tab, the display as setting must be changed to Hex[space] and half duplex must be checked off. In addition to this, under the port tab the baud must be changed to 9600, the open button must be pressed down, and the port must be changed to the port that the micro controller is connected to. Once these are done, a hexadecimal signal can successfully be sent to the micro controller. The final thing the user must do is to input a hexadecimal value of a temperature in Celsius that they wish to change the voltage regulator to.





5.2 Getting Started/How to use the device

After setting up the system and software, these are the steps you should take for using the device.

- Connect the device to a PC

  Using the usb cable that comes with the MSP430G2553, connect the micro controller to the PC.

- Set up Realterm

  Open Realterm and change the initial settings to match the settings described in the ”Communicating with the Device” section.

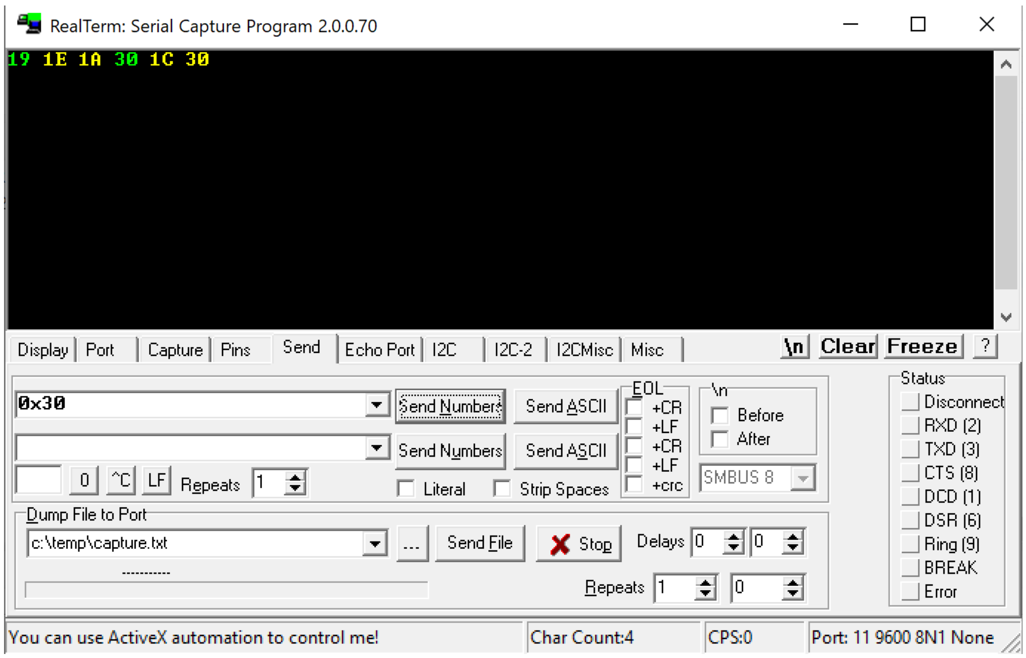
- Enter a desired temperature

  Within Realterm, send the micro controller a temperature that you would like the voltage regulator to operate at in Celsius.

- Wait for Results

  The system will take a few seconds to regulate the voltage regulators temperature to the new desired temperature and send the new current temperature.

5.3 Test Data

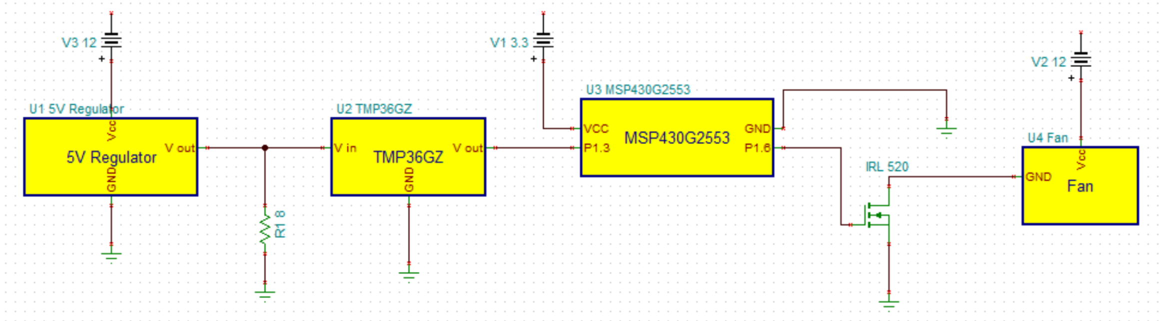


The test data above shows the system cooling and heating up. The number that is inputted is what the desired temperature that you would like. Realterm then outputs two numbers. The first number that is sent back is the temperature that the voltage regulator is read at the moment the desired temperature is inputted. The second number that was outputted was the current temperature once it is within 2 °C of the desired temperature. The reasoning for outputting the second additional number is so the user knows when the voltage regulator is at the desired temperature. The test data shows that at first the inputted number is 19 in hexadecimal which would give it a decimal value of 25 so the voltage regulator will cool down to 25 °C. The first value that is outputted is 1E which is 30 °C. The second outputted number is 1A which would make the measured temperature at 26 °C which is within 2 °C of the inputted number. This shows the system cooling down the voltage regulator to between 2 °C of the desired temperature.

Afterwards 30 is inputted into the system which means the voltage regulator should heat up to 48 °C. The first outputted number is 1C which would mean the voltage regulators current temperature is 28 °C. The second outputted number is 30 which would put the measured temperature at 48 °C which is exactly what was the desired temperature was. This successfully shows the voltage regulator heating up to within 2 °C of the desired temperature.

6 Design Files

6.1 Schematics



6.2 Bill of Materials

Item Number	Dscription	Quantity
1	MSP430G2553	1
2	AFC0712DD Fan	1
3	TMP36GZ PTAT	1
4	IRL520 NMOS	1
5	LM340T5 7805 Voltage Regulator	1
6	$8\,\Omega$ 50 W Resistor	1
7	Heatsink	2