SUPERCAPACITOR MODULE FOR ILLINI-

By

Haoyuan You

Shaurya Grover

Final Report for ECE 445, Senior Design, Spring 2023

TA: Matthew Qi

24 March 2023

Project No. 79

Abstract

The Robomaster robotics competition provides a regulation power source, with the limitation that power draw is limited to an 80W maximum. In order to provide a higher peak power for a robot, we created a supercapacitor module that can store excess power when we aren't using the full 80W, and we draw on that excess power when we want over 80W. Our solution consists of a power switching module controlled by a microcontroller receiving power requests from the robot, a charging module regulating the amount of power spent on charging the supercapacitor, and a discharging module stabilizing capacitor output to 24V. We were able to get the three subsystems working independently, but failed to connect them.

Contents

1. Introduction

Our project is to create a module to store excess power when not needed, and provide it when it is needed. We plan to achieve this using supercapacitors.

1.1 Problem

Illini-Robomaster (iRM) is an RSO at UIUC competing in the Robomaster robotics competition. During a match, robots will be penalized when motors exceed the power limit, but the monitoring system (referee system) is only checking the power output from the battery. To maximize available power for the motors and achieve greater mobility, we need a device to store and release energy. Existing solutions are either prohibited by the competition rules, too large to fit in our mobile robot, or sold at an unacceptable price by our competitor universities.

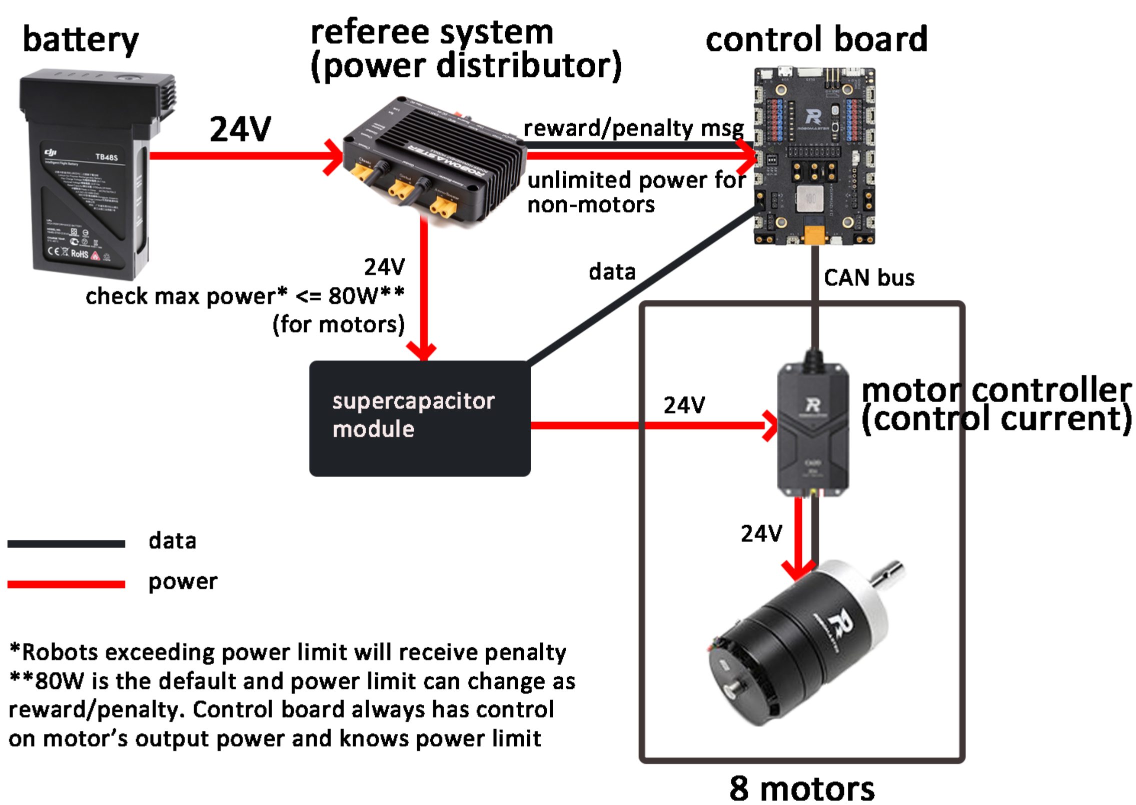

Figure 1: Visual aid. Note that the control board is the master control board on the robot, not on the module.

1.2 Solution

We propose a supercapacitor module to supply power in addition to the battery. It should be capable of storing energy from the battery when the robot is running on low power and releasing energy when the robot needs it. Thus, we have more power available.

The supercapacitor module consists of a supercapacitor array and a control module. It should keep track of the current power stored and respond to requests from the master board by changing the power pathway of motors between by the battery or by the capacitors, and changing the behavior of the capacitor between charging and discharging.

1.3 High Level Requirements

When the energy required by the motors is less than the source power limit of 80W, the module should supply the motors with enough power (as requested) and use the remaining current to charge the capacitor.

When the energy required by the motors is more than 80W and the capacitors have energy, the module should power the motors with 24V, same as the battery, and the current required by the motors.

When the energy required by the motors is more than 80W but the capacitors don’t have energy, the module should direct all power from the battery to the motor and make sure the capacitors are not charging to take up power.

2 Design

2.1 Overall Design

Using a supercapacitor as the energy storage is a method suggested by the competition manual. However, the implementation of a supercapacitor module can vary. One alternate implementation is using parallel power sources, where the battery and the capacitor can discharge at the same time so the power of the battery when the capacitor is discharging does not go to waste. This implementation is ruled out because our designs for each block were mostly focused on trying to make them as simple as possible. Using parallel power sources required precise control of the circuit or one of the power sources might burn. For our design, we chose the simplest design possible: that the charging and discharging subsystems of the capacitors are both unidirectional and cannot operate at the same time. Thus we avoid many technical issues involved in complex power system design.

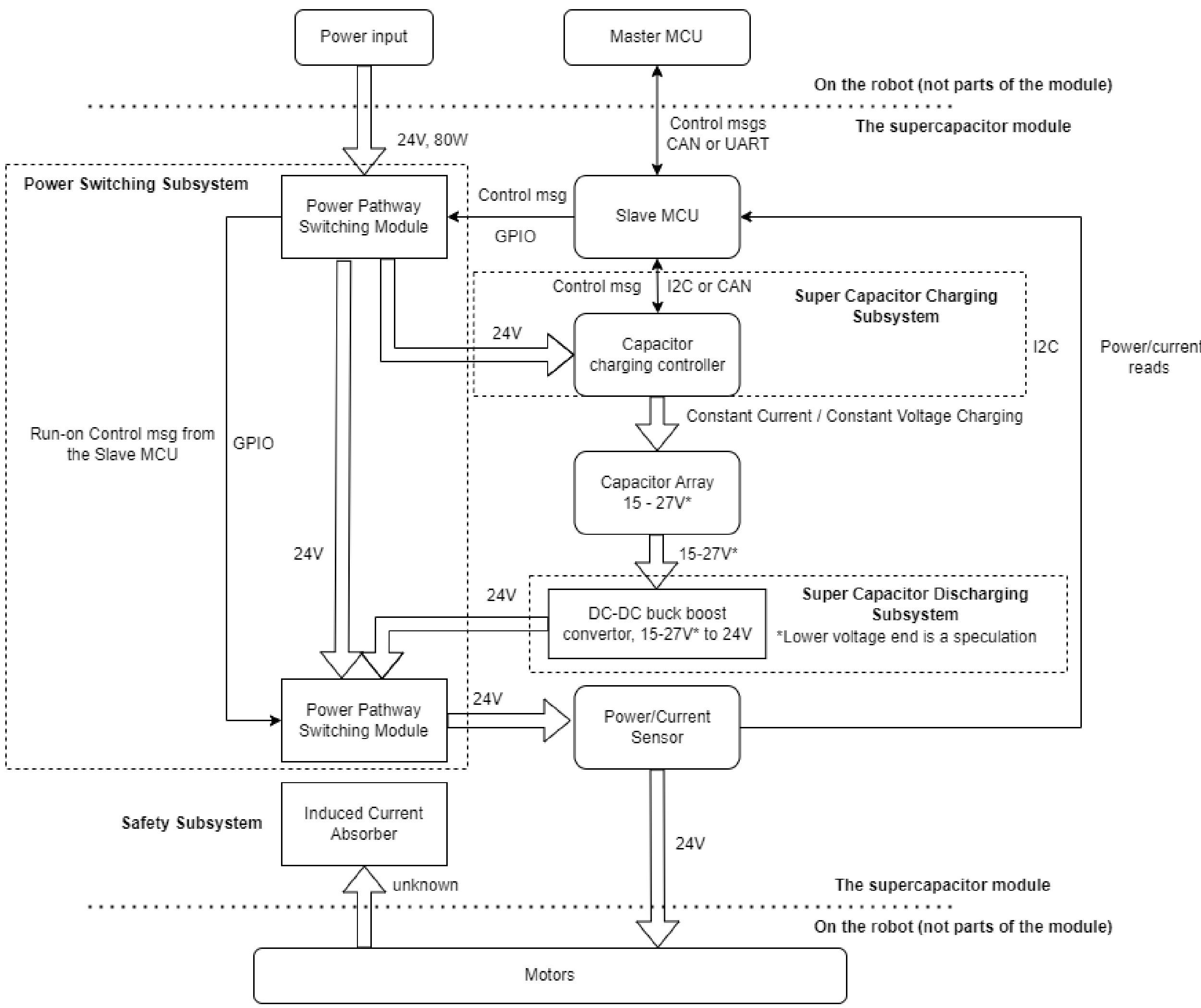

Figure 2: Block Diagram of the entire device

Legend:

Thick arrows: Power

Thin arrows: Data

Rounded Rectangles: Off-the-shelf products

Angled Rectangles: Customized circuits

2.2 Power Switching Subsystem

2.2.1 Design

The power switching subsystem switches the power pathway for the motors given the requests from the master MCU and current state.

The switching is controlled by the slave MCU. Some power pathway configuration (for example, powering motors with battery and discharging capacitors together) is prohibited and is enforced by software.

For the power switching subsystem, we use MOSFETs as switches and control their GATEs with GPIO pins on the MCU. We use the CSD19531 MOSFETs which can sustain 100V and 100A. Since the MOSFETs we chose need a 5V voltage at their gates to be activated, we apply a layer of MOSFETs controllers to take signals from the MCU and control the MOSFETs.

There are 3 different pathways that make 3 valid configurations:

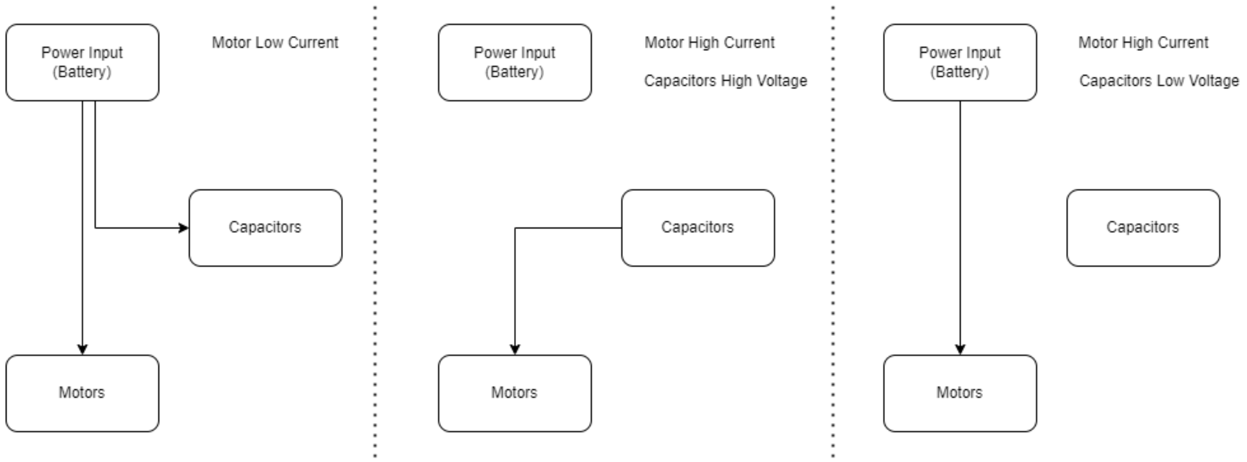

Figure 3: Power switching pathways

2.2.2 Design Decisions

For the power switching module, we chose the STM32 microcontroller because the robot has the same controller, which would make interfacing to get power requests easier. The GPIO and I2C signals the controller needs to send out are minimal, so for those purposes almost any microcontroller would work. We chose MOSFETs for our gates for their fast response and extremely high power capacity, which would satisfy our requirements. The MOSFETs need gate drivers due to their relatively high gate voltage, so we used MOSFET controllers.

2.2.3 Verification

The system was unable to be integrated due to the controller not working, but the hardware and component sections were verified separately. Applying high and low signals through the GPIO pins of the MOSFET controllers gave us the expected switching behavior, and our code logic output the correct signals.

Table: Power Switching R/V

Requirements	**Verification**
The switching system can switch to all 3 patterns of connections: battery to motors; battery to motor and battery to capacitors;	Apply threshold voltage to corresponding gates Then, measure connectivity with a multi

and capacitors to motors	scope. Confirm that only the required path is closed.
The switching system's responding time should be less than 2ms (minimal control cycle for motors)	Measure the voltage on the gates and the output with an oscilloscope Apply different voltages to the gates, and compare the time delay is < 2ms.
The switching system must sustain a large current (10A±1A)	Apply 10A current with a current source to each of the configurations for at least 10 seconds. Then go over the previous 2 tests. If both pass, then we consider this as a pass. This test is not passed because the large current will burn the wire trace of the subsystem. The components are still functional. after the large current.

2.3 Supercapacitor Charging System

2.3.1 Design

The supercapacitor charging system charges the supercapacitor with a set current from the battery. This subsystem is controlled by the slave MCU and coupled with the switching subsystem.

We choose BQ24640 as an off-the-shelf supercapacitor charging chip. It can charge the supercapacitor from 2.1V to 26V and has constant voltage and constant current charging modes. A PWM pin on the chip is used to control the charging current of the capacitor so we can control the power allocated to charge the capacitor and power the motor. Power is controlled by changing the current since the voltage on all loads is a constant 24V.

2.3.2 Design Decisions

For the supercapacitor charging module, we found a chip designed for the purpose of charging supercapacitors, which can charge with a set current or a set voltage, and stop when the required voltage is reached. We had to interface with this system via I2C, so we used the STM32 microcontroller from the power switching module for this purpose.

2.3.3 Verification

The charging chip is biased by an I2C controlled potentiometer which we lost, so we attempted to verify the system by testing with different sized resistors and measuring the power output vs the expected power output. Different resistors did change the output, but not as we expected.

Table: Supercapacitor Charging R/V

Requirements	Verifications
The charging subsystem can charge the capacitors with the set current	Set a voltage by replacing resistors at the current control pin. Then, measure the output voltage of the charging subsystem and compare it with the expected value. We tested with different resistors and the output voltage change as expected. When we replace the resistor with a piece of wire, the charging current is set to 0 and the measured output voltage is 0 as we expected.

2.4 Supercapacitor Discharging System

2.4.1 Design

This system releases energy from the battery at a stable 24V out, and is coupled with the Power switching subsystem.

For this subsystem, we will use the MP4247 non-inverting buck boost converter, which can convert a range of input voltages to a requested output voltage, in our case 24V. This chip takes I2C signals to set the output voltage and other parameters. Along with the surrounding circuitry, this will make up our discharging system, leading back to the power switching subsystem to deliver power to the robot.

2.4.2 Design Decisions

Our supercapacitor discharging module required us to convert the 15V-27V power output from the supercapacitor to a stable 24V, so we used a buck-boost converter, which can convert to both lower and higher voltages. Most buck boost converters around this power level are used for automotive applications, and have an upper limit of 21V output, so our choices were limited. We additionally needed a non-inverting converter, as conventional buck converters invert the voltage. Thus, our choice was very limited. The chip we chose also has a very simple I2C interface, only requiring one write to set an output voltage.

2.4.3 Verification

When we attempted to test this system with the input and output voltages we wanted, we found that our power trails on the PCB were too close together, and the board fried. We tested at lower voltages of 6-15V in and 5V out, and the system was successful, and voltages stayed within the +/- 0.5V tolerance.

Table: Supercapacitor Discharging R/V

Requirements	Verifications
The DC-DC converter should convert variable input voltage to stable output voltage (+/- 0.5V)	Connect a power supply to the input of the discharging subsystem. Then, scan the voltage of the power supply from 27V to 10V with a step of 0.1V Confirm that the voltage output is in the range of 24±0.5V. In our tests, the voltage stayed in 5V +/- 0.5V.
The DC-DC must sustain a large current (10A±1A)	Apply 10A current with a current source to each of the configurations for at least 10 seconds. Confirm no critical thermal effect (> 125°C) Then go over the previous test. If it pass, then we consider this as a pass.

2.5 Software

A list of complete API available is as follows (written in C):

Capacitor charging subsystem:

void SetPotentiometerValue(I2C_HandleTypeDef *mcp4018, uint8_t value)

Set potentiometer value to control the charging current

float CalcIChg(uint8_t potentiometer_value)

Compute charging current value based on set potentiometer value

float CalcPower(float IChg)

Compute power allocated to capacitor from the battery based on charging current.

Power switching subsystem:

void SetPowerSwitchMOSFET (GPIO_TypeDef* group, uint16_t pin, uint8_t state)

Set a selected MOSFET value to open or closed

void SetBQ24640CE(const int state)

Set the capacitor charger chip to be on or off

Power_Switch_State SetPowerSwitchingState(Power_Switch_State state)

Set the configuration of the entire power switching subsystem

Capacitor discharging subsystem:

void INA219ReadRegister(ina219_t *dev, uint8_t reg, uint8_t *data, uint16_t size)

Set register value for the current monitor

void INA219WriteRegister(ina219_t *dev, uint8_t reg, uint8_t *data, uint16_t size)

Read register value for the current monitor

void INA219Init(ina219_t *dev, I2C_HandleTypeDef *i2c_handle)

Initialize the current monitor

void INA219UpdateCurrent(ina219_t *dev)

Read current value from the current monitor

2.6 Tolerance Analysis

The maximum energy allowed by the rules in the capacitors is 2000J (nominal). We plan to use 10 2.7V 50F capacitors, which results in ½ CU^2 = 1822.5J at 27V. Average attacking window lasts for 5s and consumes 100W of power. So we want the capacitor module to release at least 500J of energy, and the ending voltage is 23V, which is higher than the minimum voltage our supercapacitor charging chip can support (2.1V for BQ24640). If power is drawn for longer, at a steady 100W from supercapacitors the current rapidly increases. If we draw power down to 15V in a longer attacking window, the current will go up to 6.67 A.

2.7 Safety

The only safety concern we had was dealing with the induced current from the motors when they stop. However, the robot's systems already handle this problem for us.

3. Cost and Schedule

3.1 Cost

The total cost for parts as table shown below is $59.08. With an additional $20 in shipping and 10% on tax, the final result is $84.99 We can expect a salary of $40/hour * 7 weeks * 20 hours/week = $5600 in labor cost. This comes out to be a total cost of $5684.99.

Name	Description	Price	Quantity	Total Price
SIS412DN	MOSFET for supercap charger chip	0.59	2	1.18
BQ24640	supercapacitor charge controller	5.19	1	5.19
INA219	digital current monitor	2.97	1	2.97
LM74670	smart diode rectifier controller	2.76	1	2.76
STM32G474	MCU for the board	12.35	1	12.35
MP4247	buck boost converter	3.81	1	3.81
AMS1117	Voltage regulator for MCU	0.62	1	0.62
LM5050	MOSFET controller	2.15	2	4.30
MCP4018	Digital Potentiometer	0.72	1	0.72
Capacitors	SMD 0805 capacitors	0.1 (average)	36	3.60
Inductors	SMD 0805 inductors	0.1 (average)	2	0.20
Resistors	SMD 0805 resistors	0.1 (average)	22	2.20

LED lights	indicate power status	0.15	3	0.45
SK1045	Schottky diode for supercap charging	0.85	1	0.85
BAT54	Schottky diode for supercap charging	0.16	1	0.16
SMAJ28CA	DC-DC output voltage regulator (fuze)	0.32	1	0.32
4 pin header	pinout from the MCU	0.16	6	0.96
XT30 connector	connectors for the power	1.9	4	7.60
CSD19531	MOSFET for power switching	2.21	4	8.84
Total	-	-	-	59.08

3.2 Schedule

Week	Task	Person
2.27 - 3.5	Order parts	Haoyuan
	Initial PCB design	Haoyuan
3.6 - 3.12	Verify power switching module + programming	Haoyuan
	Verify DC-DC converter module + programming	Shaurya
3.13 - 3.19	Verify safety module	Haoyuan
	Verify charging module + programming	Shaurya
3.20 - 3.26	Test the entire system	Shaurya

	Second PCB design	Haoyuan
3.27 - 4.2	Assemble	Everyone
4.3 - 4.9	Integrate to the real robot	Everyone
4.10 - 4.16	Finalize everything	Everyone

5. Conclusion

Although we failed to integrate the subsystems due to a limitation of time, we successfully have all subsystems working as expected with a testing value. We can conclude that our project has all the essential components functioning and we collected valuable data for further improvements. We will work on a second prototype to fix problems discovered during this project to reach better performance.

5.1 Accomplishments

Most of our hardware systems have worked in testing, though they have only been tested separately. Our power switching system (sans microcontroller) switches correctly when the GPIO signals are sent manually, and the software has been independently tested to work on a dev board. Our supercapacitor charging system responds to different resistances where the potentiometer would be with changes in current output, so while we cannot fully test the system it appears to at least partially work. Our discharging system similarly shows good stability for a wide range of input voltages, though we had to test it at a lower value of 5V out due to the board frying. With more time for testing and a new PCB, we could likely have all our subsystems fully working.

5.2 Failures

Our main failure is our microcontroller not working, which means we are unable to integrate any of our systems. We likely fried our only microcontroller with bad soldering, so with practice and a new chip we believe we can get it working. Our PCB layout will need to be redone focusing more on planes rather than traces for safer power flow. Additional testing will be required for our charging system, but if we fix our PCB and microcontroller, that will be the only unverified subsystem remaining.

5.3 Future Work

As this project is for a real robot for the university's Illini Robomaster team, we plan to continue working on it. Our next steps will be ordering more parts and redesigning our PCB, followed by another round of testing. We could consider a simpler microcontroller for easier soldering, but otherwise we plan to keep our design as it currently is.

6. Ethics and Safety

As the use of supercapacitors is allowed and already regulated by rules in the competition, there are no known ethical concerns about using such a module on the robot. The safety concerns mostly revolve around storing large amounts of power in our module, as all the surrounding power sources are already determined safe and provided to us by Robomaster. We are limiting the capacitor voltage to reduce the danger present, and only charging the capacitor to a safe amount of energy stored, as ruled by Robomaster. Induced current from the motors can also damage our hardware, so we will have a subsystem dedicated to dealing with it.

Additional safety concerns exist, not during the use of this device, but during our testing. Precautions need to be taken to keep ourselves safe and to confirm the power stored in our device is no more than expected. We intend to use the proper lab equipment and safety procedures, and monitor the power stored in the supercapacitors at all times. We also intend to start testing with lower power to stay safe and ensure everything functions properly.

7. References

[1] *IEEE Code of Ethics,* IEEE Xplore, accessed 29 March 2023.