Comprehensive Design Review

Solar Winds - Team 6

March 5th, 2019

Faculty Advisor: Stephen Frasier

Meet the Team

Ajey Pandey
EE

Jayme Gordon
EE

Richard Kornitsky
EE

Jason Sproviero
EE

Nicholas McCarthy
EE

Outline

- Review of Project
  - Problem Statement & Solution
  - Use Case: County Fair
  - System Overview
    - Block Diagram
    - Individual Blocks
- CDR Deliverables
  - Demonstration at end
- FPR Goals & Plan to Achieve
  - Table of Requirements & Specifications
  - Proposed FPR Deliverables
  - Gantt Chart
  - Individual Responsibilities
- Demonstration

Problem Statement

- Cooling draws a significant amount of power
- Cooling is a "peaky" load
  - Daily, cooling peaks around 3-4PM
  - Annually, cooling peaks in late summer
- Solar power makes peaky loads even peakier
  - Phenomenon known as the "duck curve"
- Grid-scale battery storage is in research phase
- Industry focus has been storing electrical (solar) power as electrical energy

Our Solution: Solar Winds

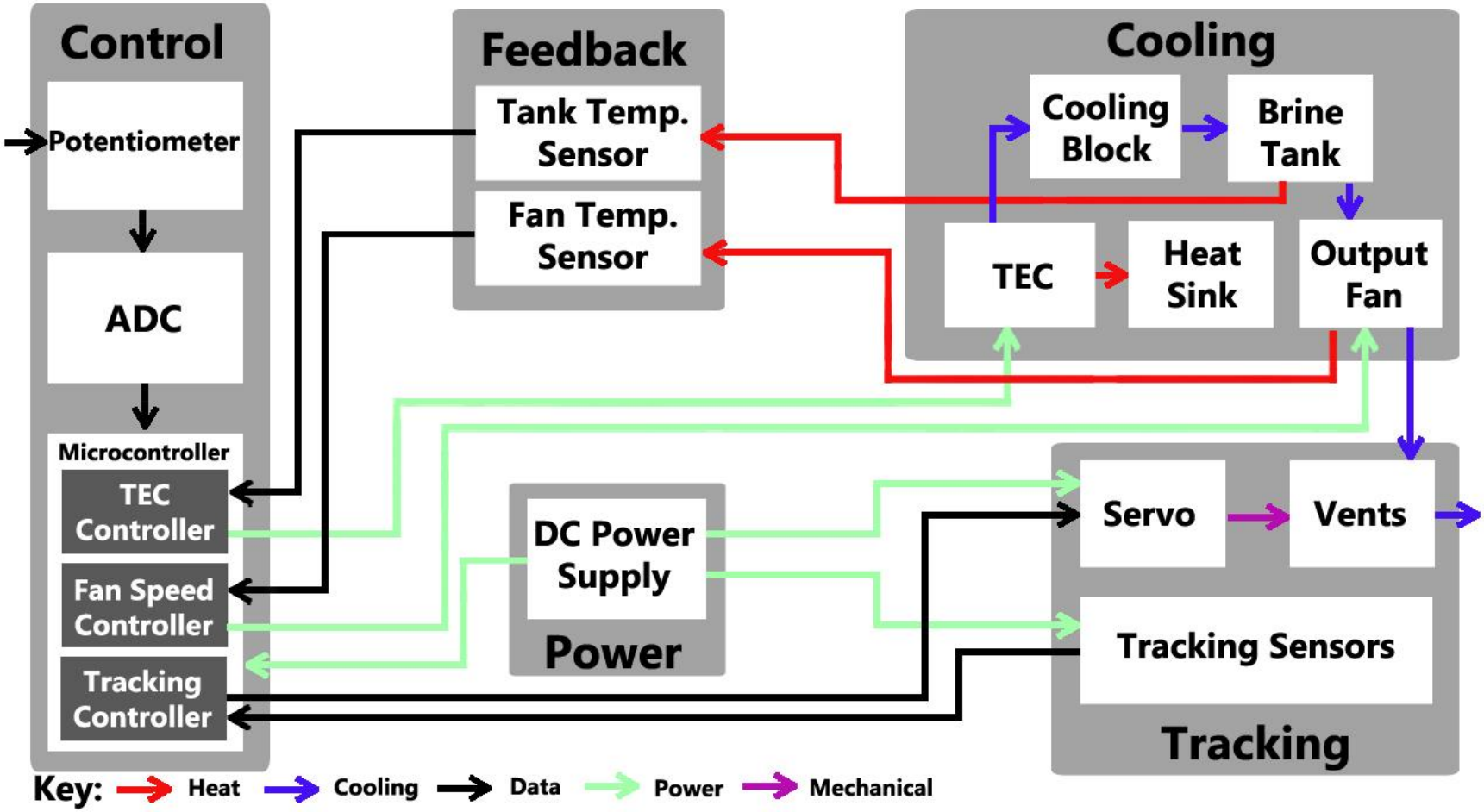
Store electrical power as thermal energy during non-peak load times

- Develop contained cooling system
- Store thermal energy in liquid solution
- Deliver cooling with stored energy

Use Case: County Fair

- Temperature is ~ 80° F
- User has a van or light-duty pickup truck
- Cooling system distributes air
- Noticeably cool small volume by ~ 3.5° F

System Overview: Block Diagram



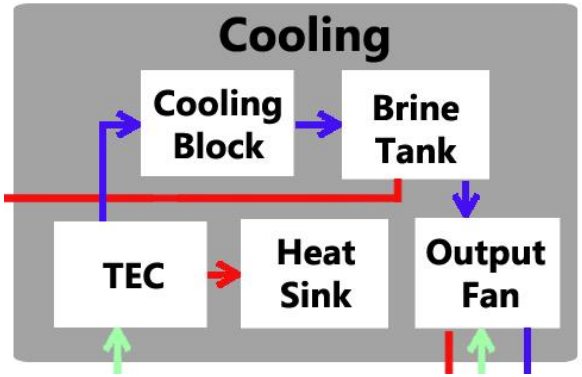
System Overview: Implementation of each Block

Cooling Block

Objective:

Circulate brine through pumping system to cooling block

Parts:



Thermoelectric Cooler: Laird HiTemp 387001828 (ET series)

Chosen operating conditions: 24V, 192W

Maximum operating conditions: 46V, 300W

Cooling block: DIYhz Aluminum Radiator (40mm x 40mm x12mm)

Liquid Cooler Heat Sink: CORSAIR Hydro Series H60 120mm

Control Block

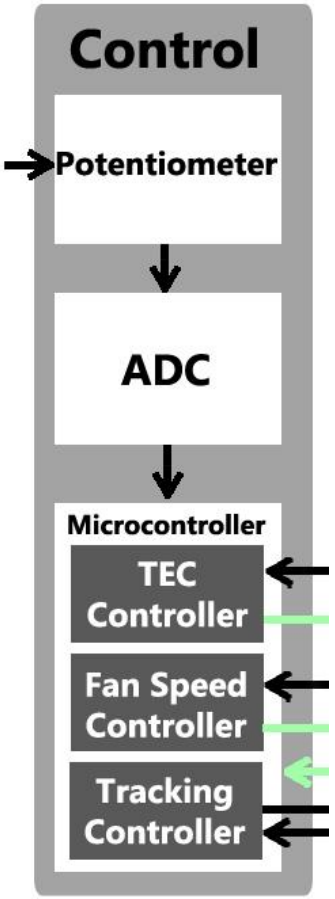
Objective:

User control of cooling

Parts:

User Input: Control circuitry

Microcontroller: ATmega328



Feedback Block

Objective:

Report data from cooling system to microcontroller

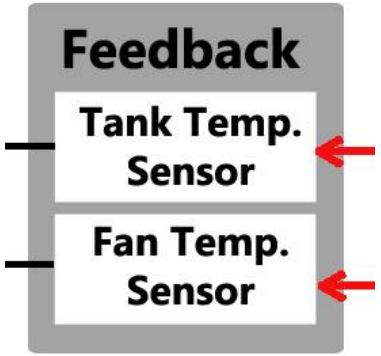
Parts:

Tank temperature sensor: DS12B20 integrated circuit

Selection criteria: Waterproof

Fan temperature sensor: LM35 integrated circuit

Selection criteria: Readily available



Tracking Block

Objective:

Develop array of sensors to

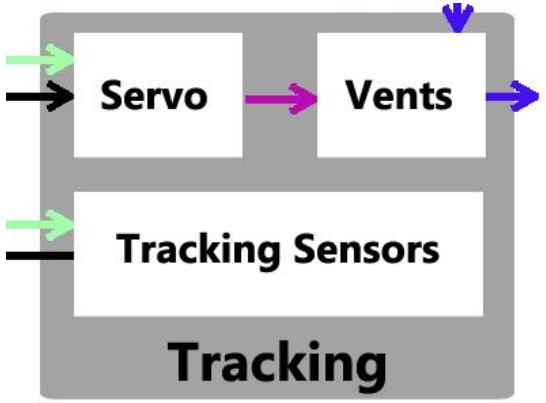
track user

Parts:

Servomotor: Direct vents towards user

Interrupt: Allow user to manually interrupt

Sensors: Detect user motion



Power Block

Objective:

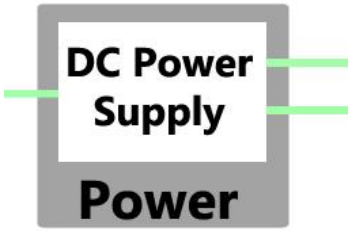
Provide power to each block

Parts:

Power Supply:

Chosen operating condition: 24V

Maximum operating conditions: 24V, 20A, 480W



CDR Deliverables

- Demonstration of Complete System Functionality
  - Build custom cooler box - Jayme & Richard
  - Data collection tool - Jayme & Nick
  - Feedback loop & temperature control - Nick & Ajey
  - Printed circuit board (PCB) - Jason
  - Sensors for directionality & prediction - Ajey & Richard

CDR Deliverables

- Demonstration of Complete System Functionality
  - ✔ Build custom cooler box - Jayme & Richard
  - ✔ Data collection tool - Jayme & Nick
  - ✔ Feedback loop & temperature control - Nick & Ajey
  - ✔ Printed circuit board (PCB) - Jason
  - ✔ Sensors for directionality & prediction - Ajey & Richard

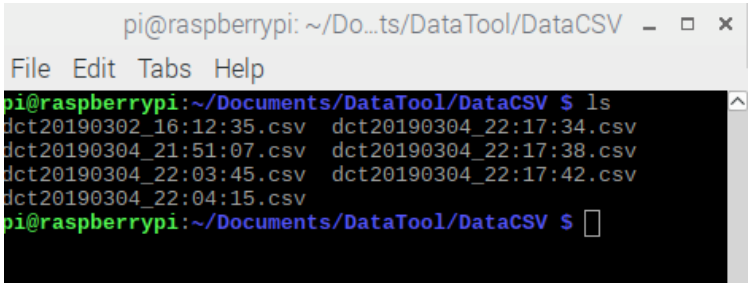
Outline of Demonstration

Build Custom Cooler Box - Jayme & Richard

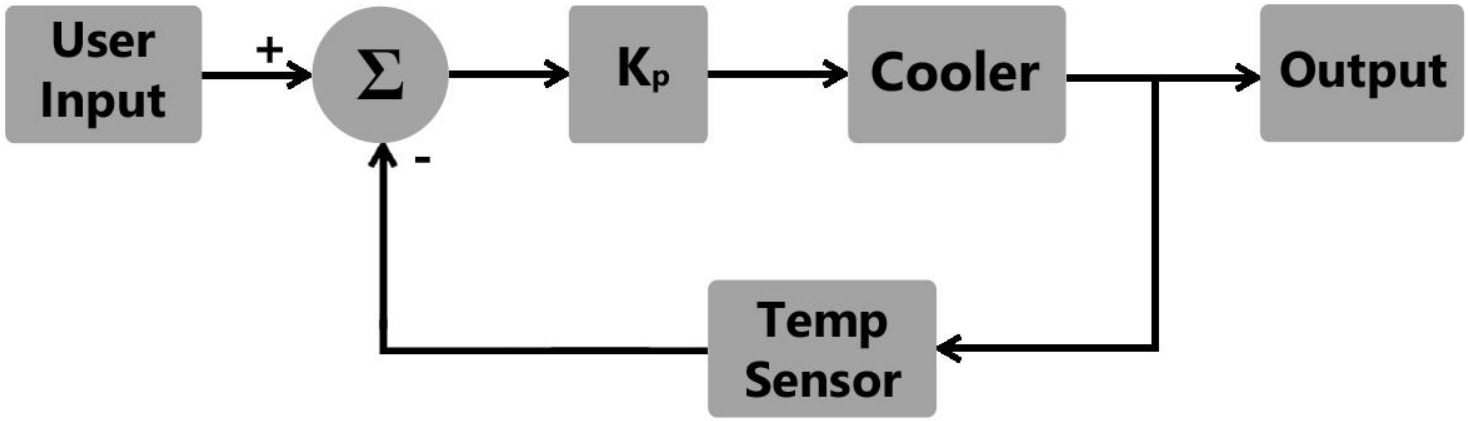
Data Collection Tool - Jayme & Nick

```
import spidev
import time
import os
import glob
from datetime import date, datetime
import RPi.GPIO as GPIO       ## Import GPIO library

GPIO.setmode(GPIO.BOARD)      ## Use board pin numbering
GPIO.setup(37, GPIO.OUT)      ## Setup GPIO Pin 11 to OUT
...
startup = 0
while startup < 5:
        GPIO.output(37,True)  ## Turn on Led
        time.sleep(1)         ## Wait for one second
        GPIO.output(37,False) ## Turn off Led
        time.sleep(1)         ## Wait for one second
        startup = startup + 1
...
# initialize GPIO pins for digital sensor
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

# Finds the correct device file that holds the temperature data
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

# A function that reads the sensors data
def read_temp_raw():
  f = open(device_file, 'r') # Opens the temperature device file
  lines = f.readlines() # Returns the text
  f.close()
  return lines

# Convert the value of the sensor into a temperature for digital
def read_temp():
  lines = read_temp_raw() # Read the temperature 'device file'

  # While the first line does not contain 'YES', wait for 0.2s
  # and then read the device file again.
  while lines[0].strip()[-3:] != 'YES':
    time.sleep(0.2)
    lines = read_temp_raw()

  equals_pos = lines[1].find('t=')

  if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c
```

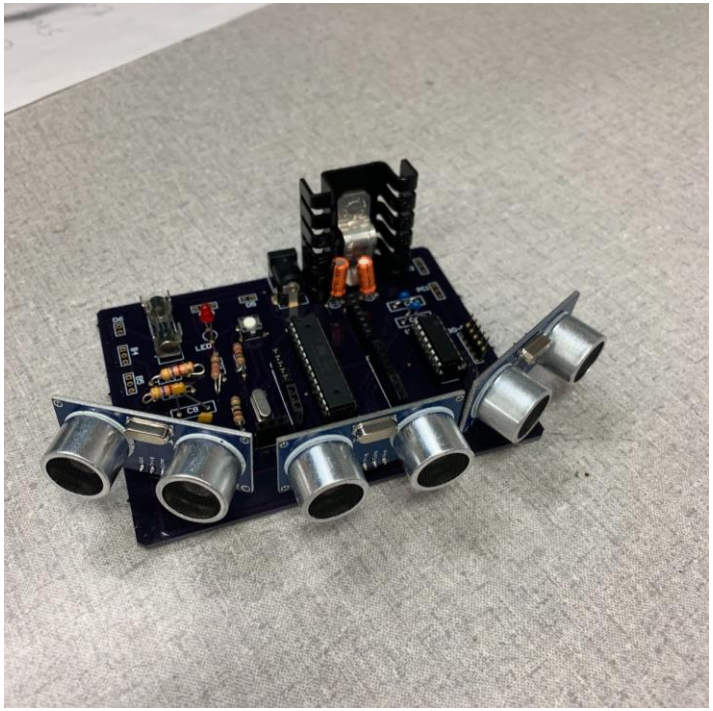
...

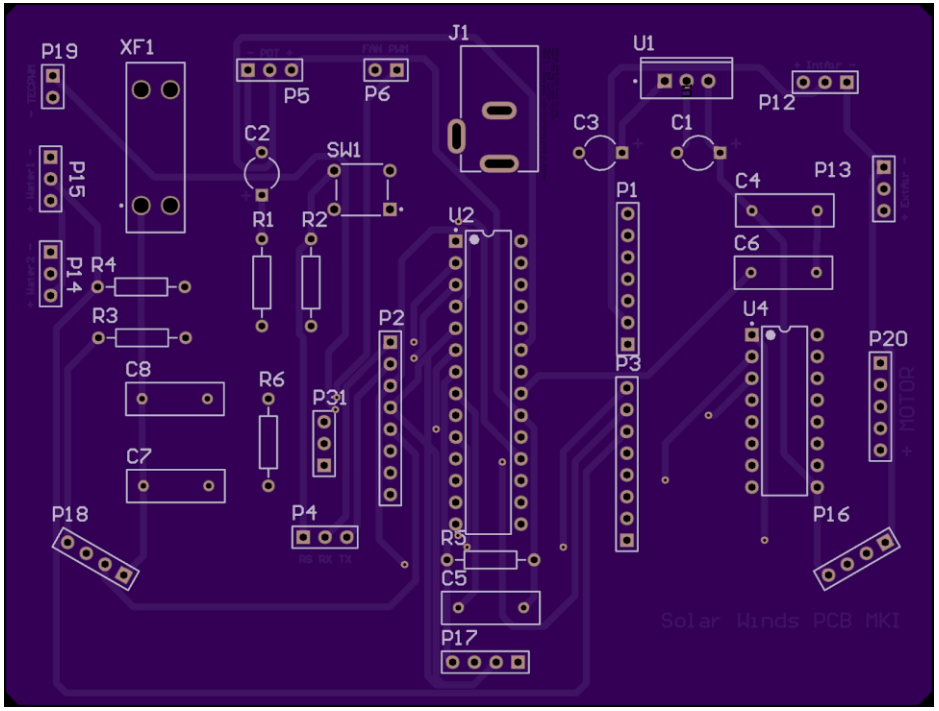


Feedback Loop & Temperature Control - Nick & Ajey

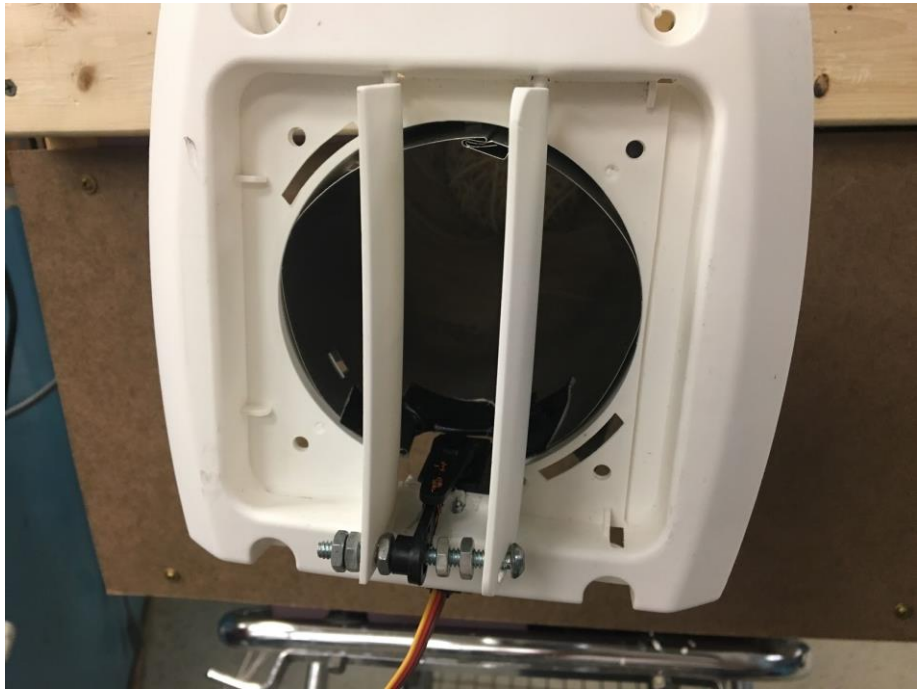


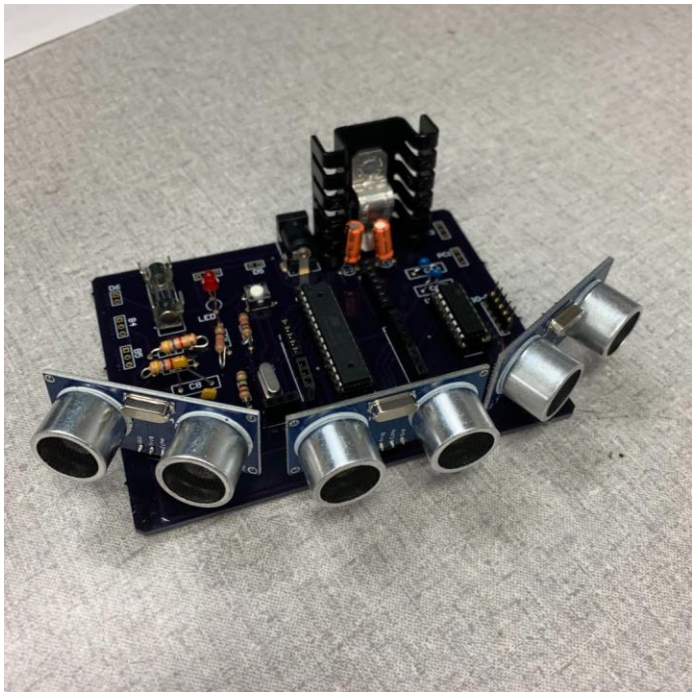
Printed circuit board (PCB) - Jason





Sensors for Directionality & Prediction - Ajey & Richard





Goals for FPR

Finished Functioning System meeting Table of Requirements & Specifications:

- Refine data collection tool
- Refine directionality
- Refine feedback
- Program in C
- Refine PCB
- Test refined PCB
- Graphical data presentation tool
- Power flow analysis efficiency report
- FPR Documentation
- FPR Preparation
- FPR Presentation
- SDP Demo Day

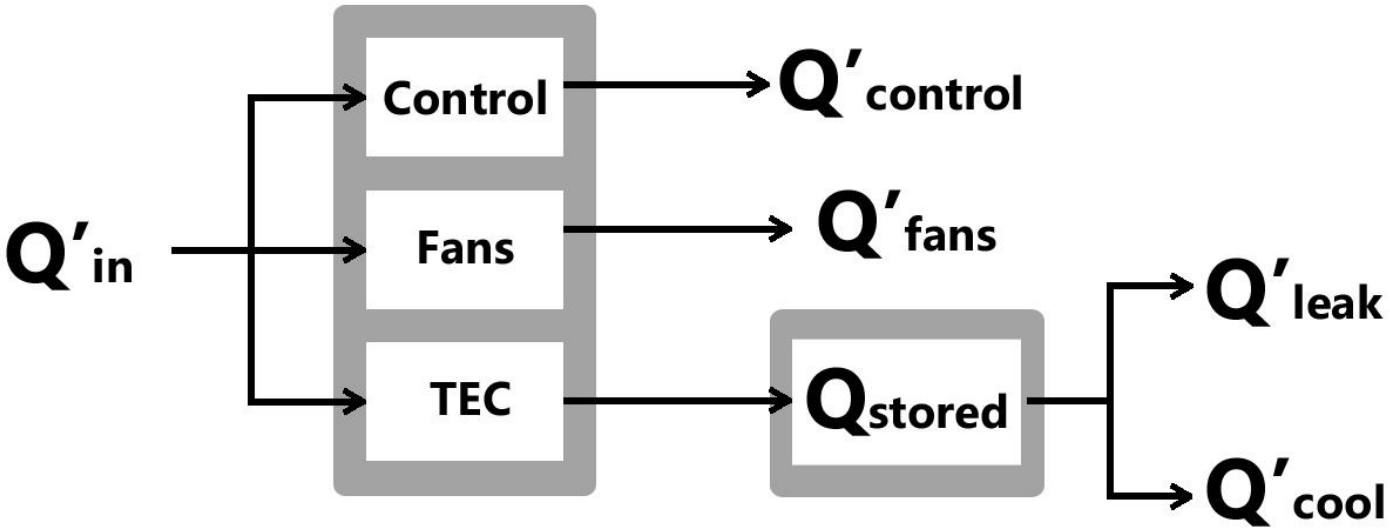
Requirement	Specification	Value
portable	dry weight	< 150 lb
	size	< 20 cuft
cooling	air cooling	3.5° F
responsive	water cooling	< 6 hours

Plans to Achieve FPR Goals

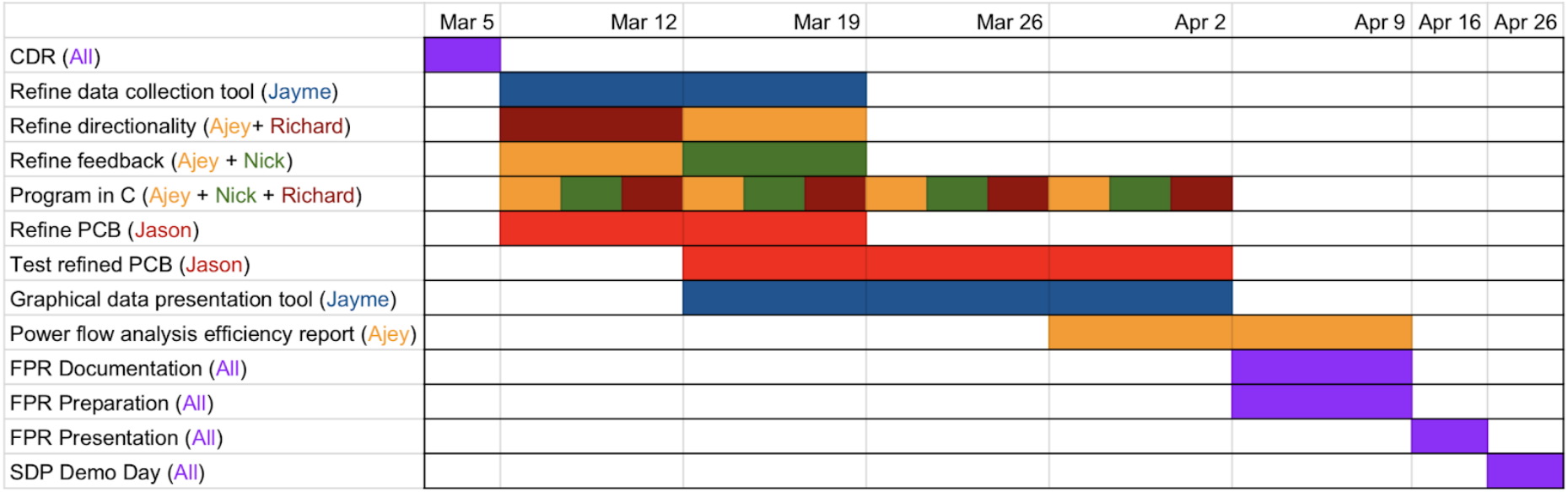
Proposed FPR Deliverables

- Refine Complete System Functionality
  - Refine data collection tool - Jayme
  - Test & optimize feedback controllers - Nick & Ajey
  - Refine printed circuit board (PCB) - Jason
  - Refine directionality - Ajey & Richard
- Added Goals
  - Graphical data presentation tool - Jayme
  - Power flow analysis efficiency report - Ajey

Mapping of Power Flow



Gantt Chart


Mar 5
Mar 12
Mar 19
Mar 26
Apr 2
Apr 9
Apr 16
Apr 26
CDR (All)
Refine data collection tool (Jayme)
Refine directionality (Ajey+ Richard)
Refine feedback (Ajey + Nick)
Program in C (Ajey + Nick + Richard)
Refine PCB (Jason)
Test refined PCB (Jason)
Graphical data presentation tool (Jayme)
Power flow analysis efficiency report (Ajey)
FPR Documentation (All)
FPR Preparation (All)
FPR Presentation (All)
SDP Demo Day (All)

Individual Responsibilities

Ajey - Sensors for directionality & prediction, program feedback loop, normalize temperature sensors, add user input, heat transfer analysis & testing, *Power flow analysis efficiency report*

Jason - Protoboard & PCB for microcontroller

Jayme - Custom cooler box built, data collection tool, *Graphical data presentation tool*

Nick - Program feedback loop, normalize temperature sensors, add user input, ~~data collection tool~~

Richard - Sensors for directionality & prediction, custom cooler box built, heat transfer analysis & testing

Note: Strikethrough to remove responsibility, Italicization to add responsibility

Thank You

Demonstration