CELL-U-LATER

ELCT 404 Final Presentation

Kevin Crean, Hudson Dye, Wyatt Hill, Davis Hobbs

AGENDA

- Project Overview
- Team Organization
- Design Philosophy
- Project Status

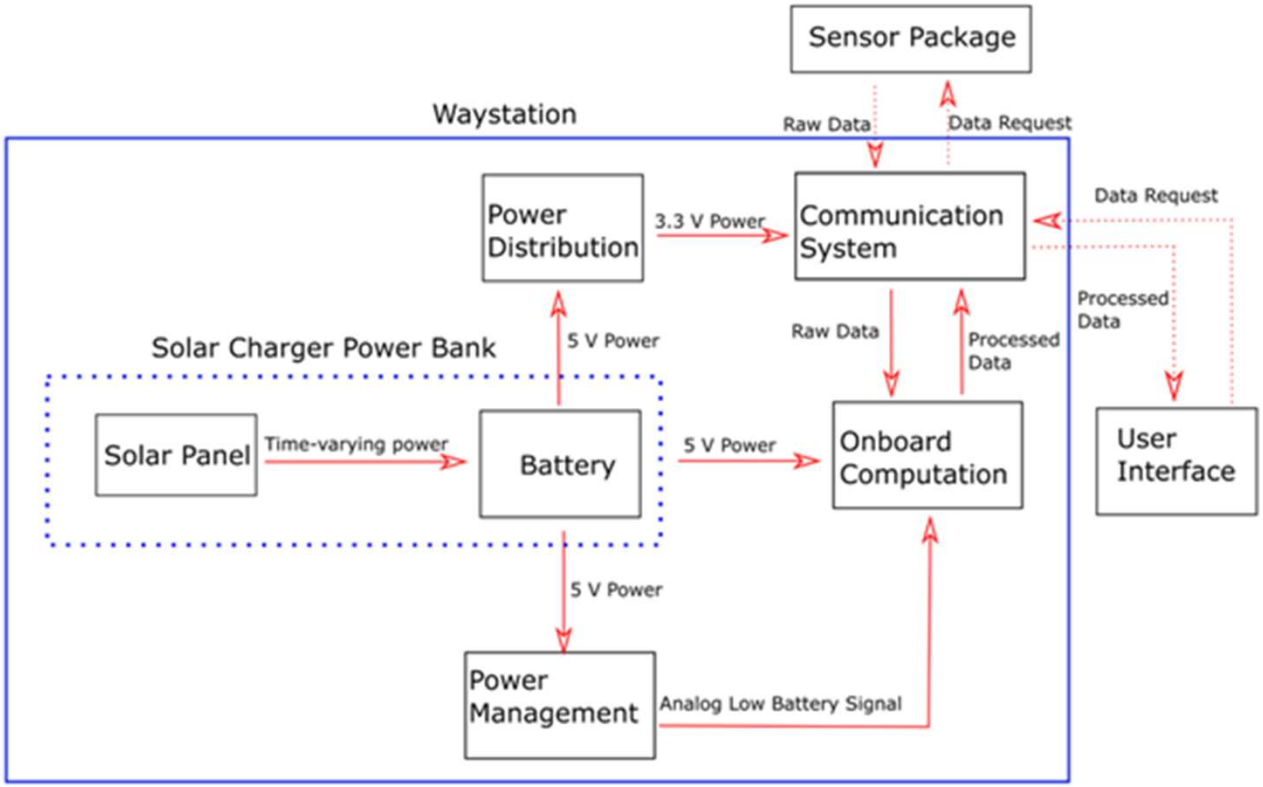
PROJECT OVERVIEW

Project Mission: Create a data collection waystation that collects data from deployed sensor packages, computes relevant metrics, and sends the metrics through a commercial cellular network.

Project Description:

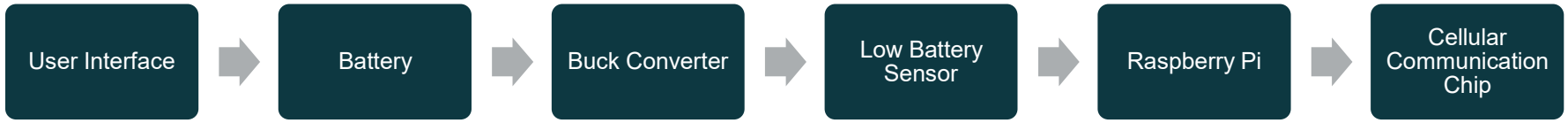
- Wireless communication between the waystation and the sensors.
- Onboard computing at waystation to calculate relevant metrics to reduce data transfer.
- Results passed on to a host PC via commercial cellular network

SYSTEM BLOCK DIAGRAM



DESIGN PHILOSOPHY

- Open-Source Project: Reduce Cost
- Accessible Implementation
- Build on Existing Methods for Sensor-to-Sensor Communication



TEAM ORGANIZATION

- Project Sponsor: Austin Downey
- Team Lead: Kevin Crean

- Onboard Processing: Kevin Crean
- Wireless Communication: Hudson Dye
- Power Distribution and Management: Davis Hobbs
- User Interface: Wyatt Hill

FINANCIALS

Component	Price
Botletics SIM700A-LTE-Shield (2)	$48
Micro SIM card + IoT subscription	$24
LP311P Differential Comparator (2)	$3
OKR-T-3-W12-C Buck Converter (2)	$16
SUNER BC-10 Solar Panel	$56
Talent Cell 5V/12000mAh Lithium-ion Battery	$40
SAE– Barrel Jack adapter	$10
Raspberry Pi 4 Model B 4GB	$106
Solar Charger Power Bank, 60000mAh Portable Charger	$56
	$353

ACHIEVEMENTS

- Single Data Point wireless communication between Raspberry Pi and Teensy
- Reliable 3.3 V regulation for pulsed power loads
- Onboard Computation of CSV file (Mean, Standard Deviation, FFT)
- Access Cellular Infrastructure

HIGH LEVEL PROBLEMS

- Wireless transmission of CSV file between Raspberry Pi and Teensy
- Communication between Raspberry Pi and Botletics SIM7000
- Implementation of the User Interface
- Complete integration of subsystems

ONBOARD PROCESSING

Kevin Crean

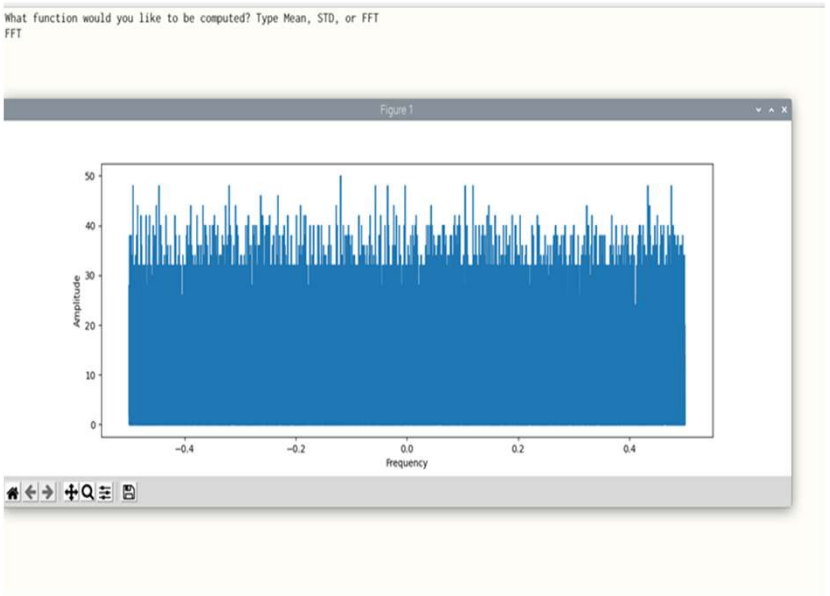
HARDWARE AND SOFTWARE REQUIREMENTS

- Raspberry Pi 4 Model B

- Python Programming Language
  - Pandas: Data Analysis library
  - Numpy: Scientific Computations
  - Matplotlib: Visualizations

ONBOARD COMPUTATION

- Read Excel File
- User requests for their computation
  - Mean, Standard Deviation, Fast Fourier Transform
- Request computation printed to serial monitor



222,221 data points

ONBOARD COMPUTATION

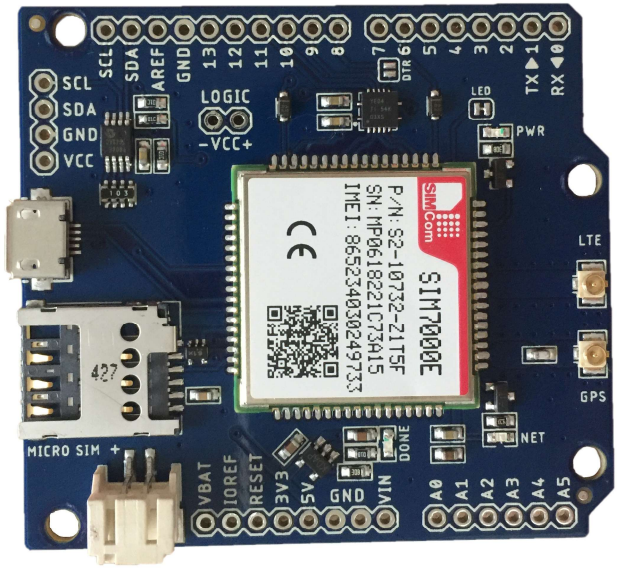
Test	Requirements
Read Function	Read excel file of 50,000 points and printed the results.
Fast Fourier Transform Function	Perform Fast Fourier Transform of excel file and print a graph of the frequency and amplitude of each data point.
Mean Function	Add all data point values to create an average value.

CELLULAR COMMUNICATIONS

Hudson Dye

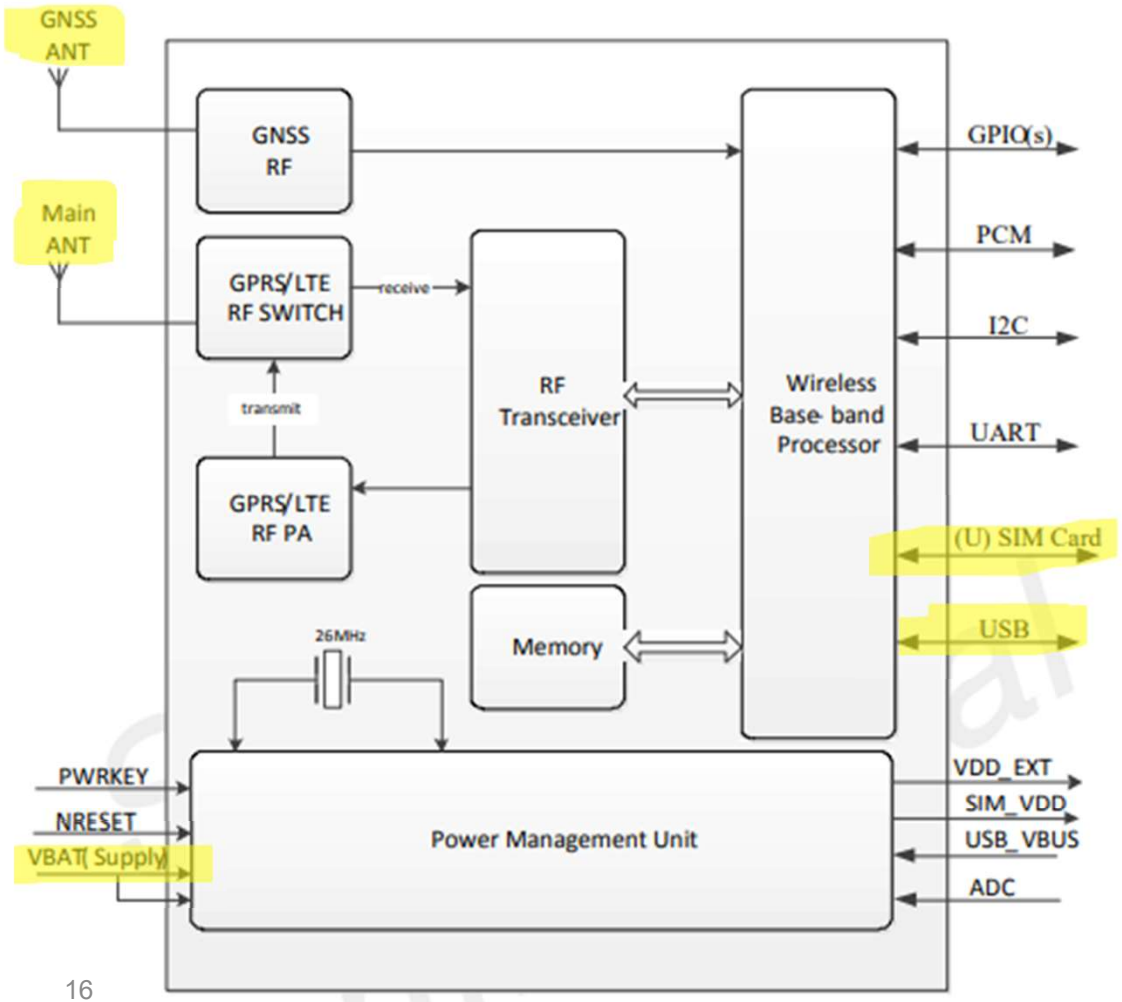
OVERVIEW: SIM7000

- Wired Control
  - Interface via desktop and pi
    - SPI, USB, etc.
- Wireless Access
  - Access through Hologram
  - Access through cell phone

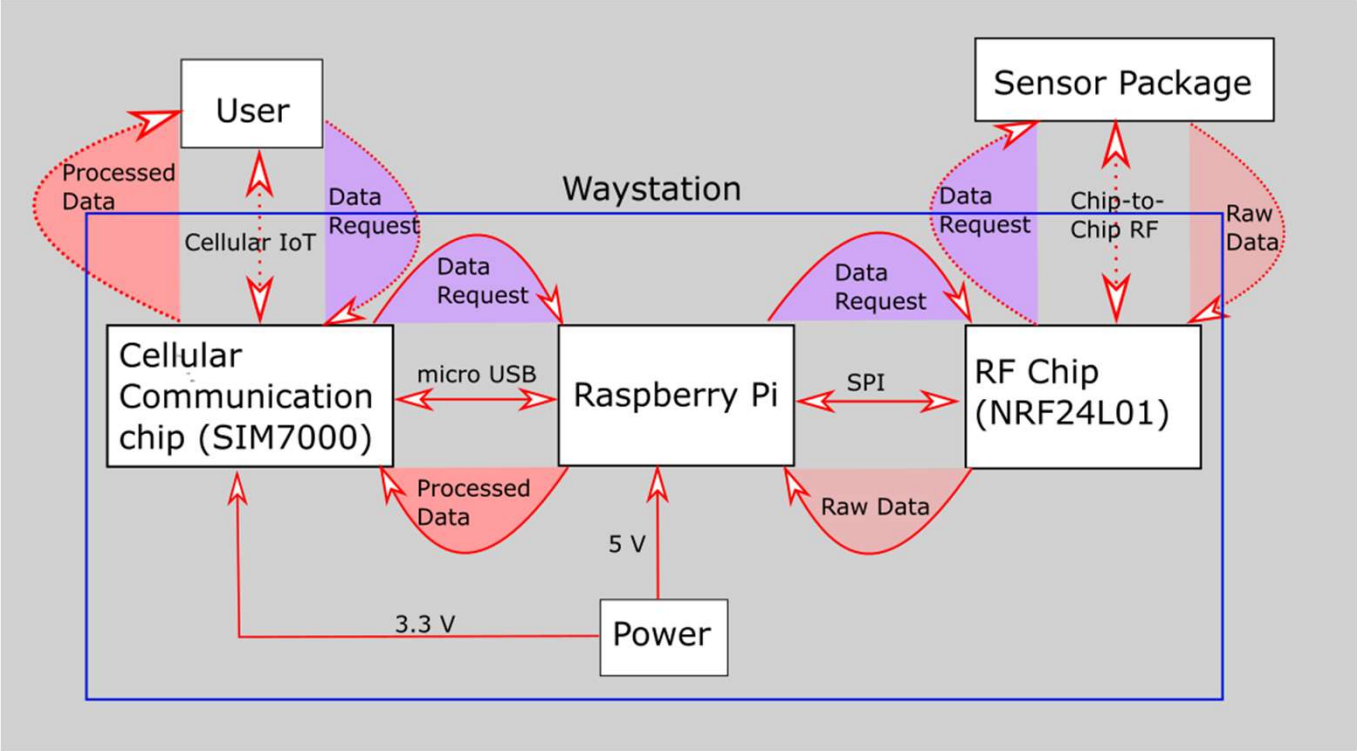


Engineering and Computing

OVERVIEW: SIM7000



WIRELESS COMMUNICATIONS



- Information flow:
  - Purple – request
  - Red – data

ARDUINO TO SIM

- Arduino Shield
- Control via with IDE via USB type-B
- Requires extra current supplied by battery (3 - 4.3 V)
- Wireless access requires SIM card and IoT subscription

SIM7000 FUNCTIONS

- Read chip & source information
  - battery voltage, ID, etc.
- Read wireless environment information
  - registration status, signal strength, network time, etc.
- Send & Receive SMS
- Post chip information to dweet.io

PROCESS

- Use Arduino Uno for prototyping
  - 101 not compatible, uno was used
- Spent weeks testing faulty chip, delaying RF communication

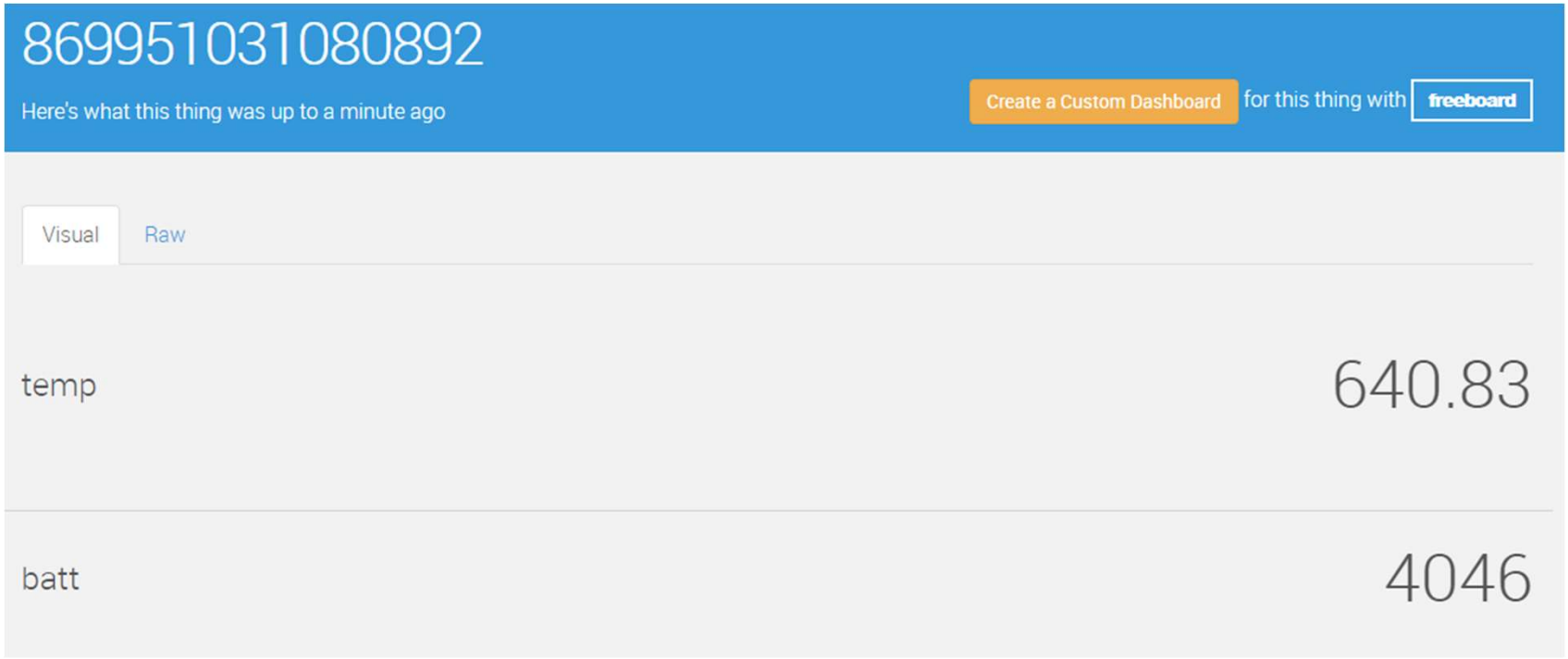
TROUBLESHOOTING

- Symptom: No network access
  - Cause: SIM7000A did not acknowledge inserted SIM card
  - Identified by: unable to read SIM card ID (CCID)
  - Helpful methods:
    - Serial tube
    - Echo
  - Solution: Replace cellular chip

AT COMMANDS

Command	Description
A/	Re-issues the last command given
ATD	Mobile originated call to dial a number
ATE	Set command echo mode
ATH	Disconnect existing connection
ATI	Display product identification information
ATL	Set monitor speaker loudness
ATM	Set monitor speaker mode
+++	Switch from data mode or ppp online mode to command mode
ATO	Switch from command mode to data mode
ATQ	Set result code presentation mode
ATS0	Set number of rings before automatically answering the call
ATS3	Set command line termination character
ATS4	Set response formatting character
ATS5	Set command line editing character
ATS6	Pause before blind dialing
ATS7	Set number of seconds to wait for connection completion
ATS8	Set number of seconds to wait for comma dial modifier encountered in dial string of D command
ATS10	Set disconnect delay after indicating the absence of data carrier

CELLULAR CHARACTERIZATION: DWEET.IO



ASSESSING GOALS

- All basic functionality is a success
  - Send/receive wired
  - Send/receive wireless
    - SMS
    - To website
- Needs work: Replace Arduino with Pi

GIVEN MORE TIME:

- Control method: SMS messages from personal cell phone
- Inputs: waystation phone number, desire node(s), & desired function
- Output: determined by desired function input
  - Ideally, data automatiaclly uploaded to central location

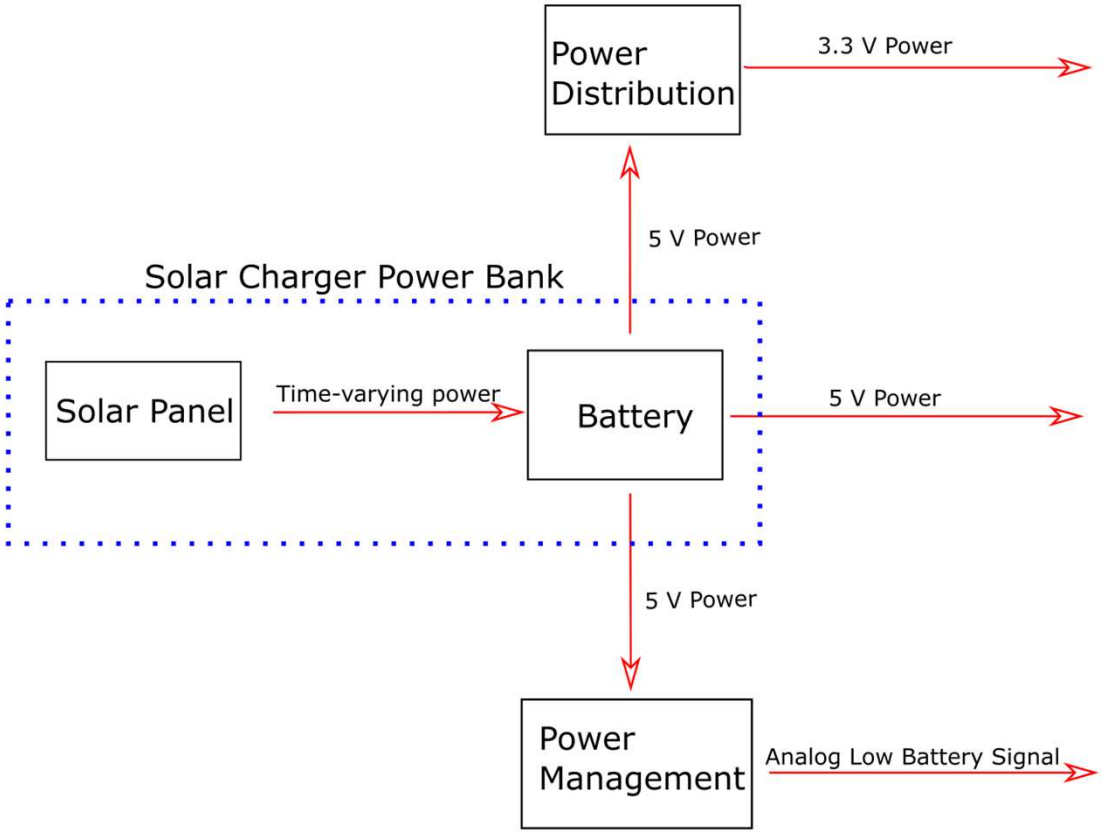
POWER AND CHIP-TO-CHIP RF

Davis Hobbs

OVERVIEW

- PDM Subsystem Hardware and Validation
  - Power Supply
  - Power Distribution
  - Power Management
- Chip-to-Chip RF
  - Teensy-to-Teensy Communication
  - Pi-to-Teensy Communication
  - Future Work

PDM SUBSYSTEM OVERVIEW



POWER SUPPLY HARDWARE

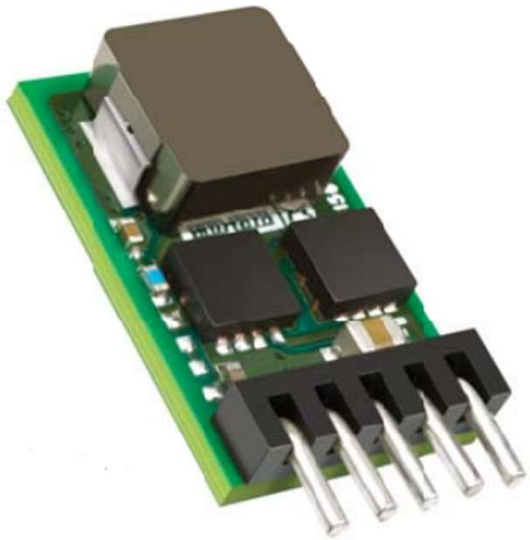
- Final Product uses PSOO Power Bank
  - Changed from 5 V Talent Cell Battery
- 222 Whr Capacity at 5 V
  - Supplies Pi at full load for 4.5 hours
- Integrated Solar Panel
- External LED Capacity Indicators

POWER SUPPLY VALIDATION

- All thresholds but capacity met
  - Exceed current threshold by 100 mA
  - Solar recharging validated
- Capacity far less than advertised
  - Expected capacity of 222 Whr
  - Measured capacity of < 215 Whr via discharge testing

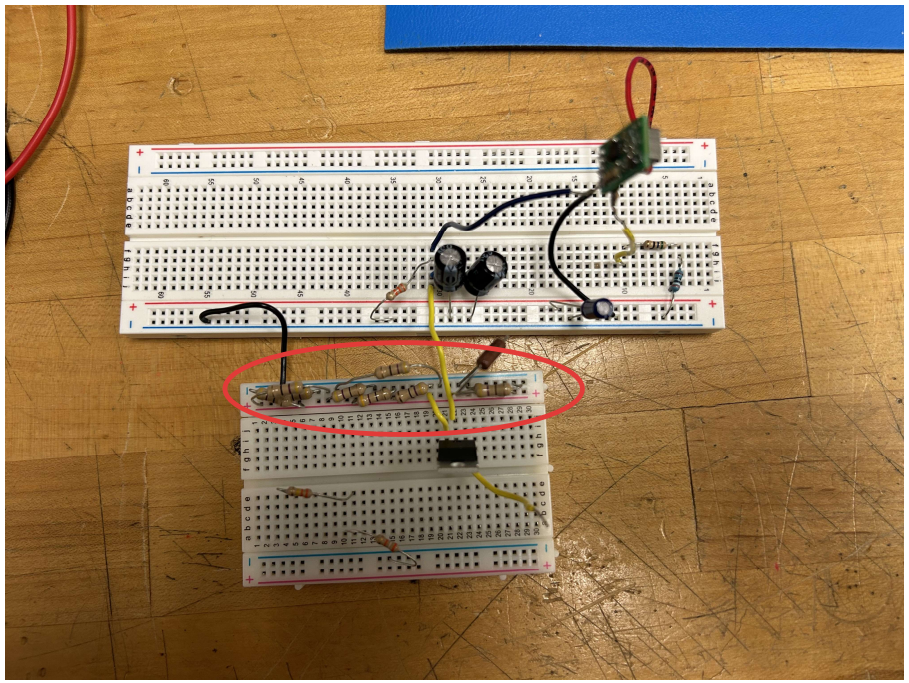
POWER DISTRIBUTION HARDWARE

- OKR buck converter supplies power to SIM7000 chip
  - Reduces ~5V battery voltage to ~3.3V
  - Up to 93% efficient power conversion
  - Highly effective at responding to pulsed loads



POWER DISTRIBUTION VALIDATION

- Thresholds met after testing dynamic load response
  - A 4.4 Ω load was toggled at a 10 Hz frequency
  - Exceeded dynamic load current threshold by 5 mA at 3.28 V–3.38 V
- Testing was far more demanding than cellular load

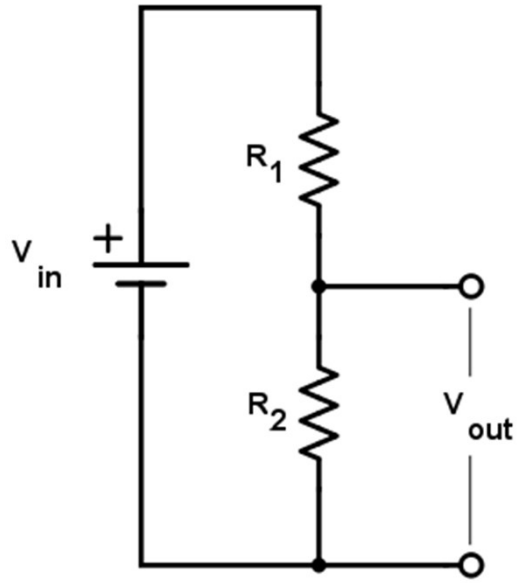


POWER MANAGEMENT HARDWARE

- Sensor implemented passively with a light-dependent resistor (LDR)
  - LDR has negligible resistance exposed to a bright light
  - LDR has resistance above 600 kOhm in a dark environment
- An LDR attached to the 25-50% charge indicator tracks battery charge
  - Capacity >= 25%: Light is bright
  - Capacity < 25%: Light is off

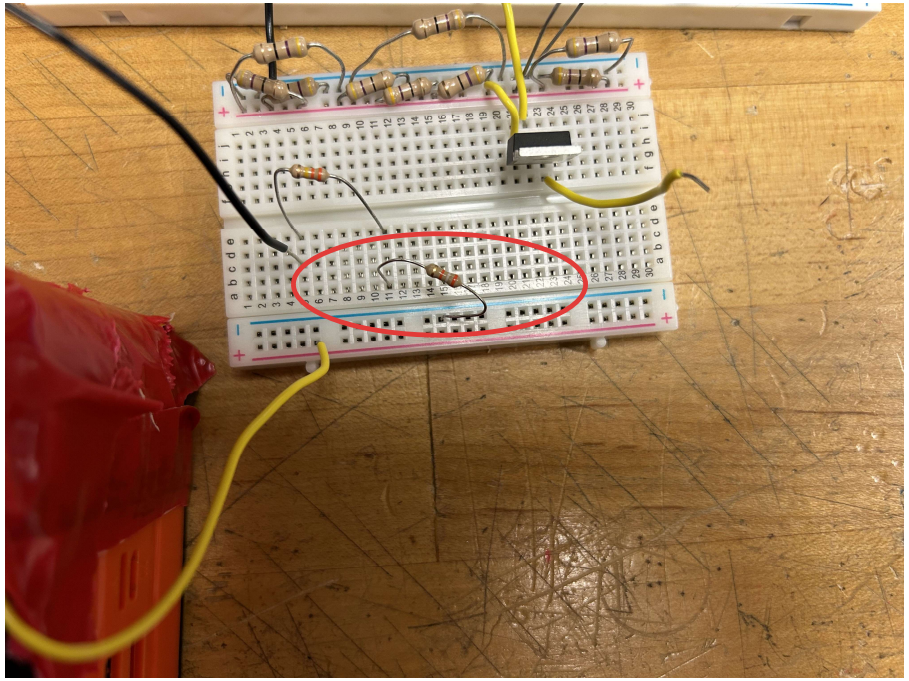
POWER MANAGEMENT CIRCUIT

- The sensor is implemented via an LDR
  - R1: LDR in series with 330 kΩ resistor
  - R2: 220 kΩ resistor
- The voltage on R2 will be used to interrupt the Pi when capacity falls below 25 %



POWER MANAGEMENT VALIDATION

- Voltage on R2 mirrored LED status in 10 trials
- The Pi has a hardware interrupt value of 1.8 V
  - LED off: 1.51 V average
  - LED on: 2.06 V average
- Sensor will likely be more accurate if encased in solid container



CONCLUSION FOR PDM SUBSYSTEM

- All threshold values in engineer's control were met
  - Battery capacity requirement failed due to false advertising
- A new battery may be needed to meet uptime requirements
  - Deficit in capacity may be offset by solar recharging
- Power management may be scrapped to decrease capacity shortfall

CHIP-TO-CHIP RF

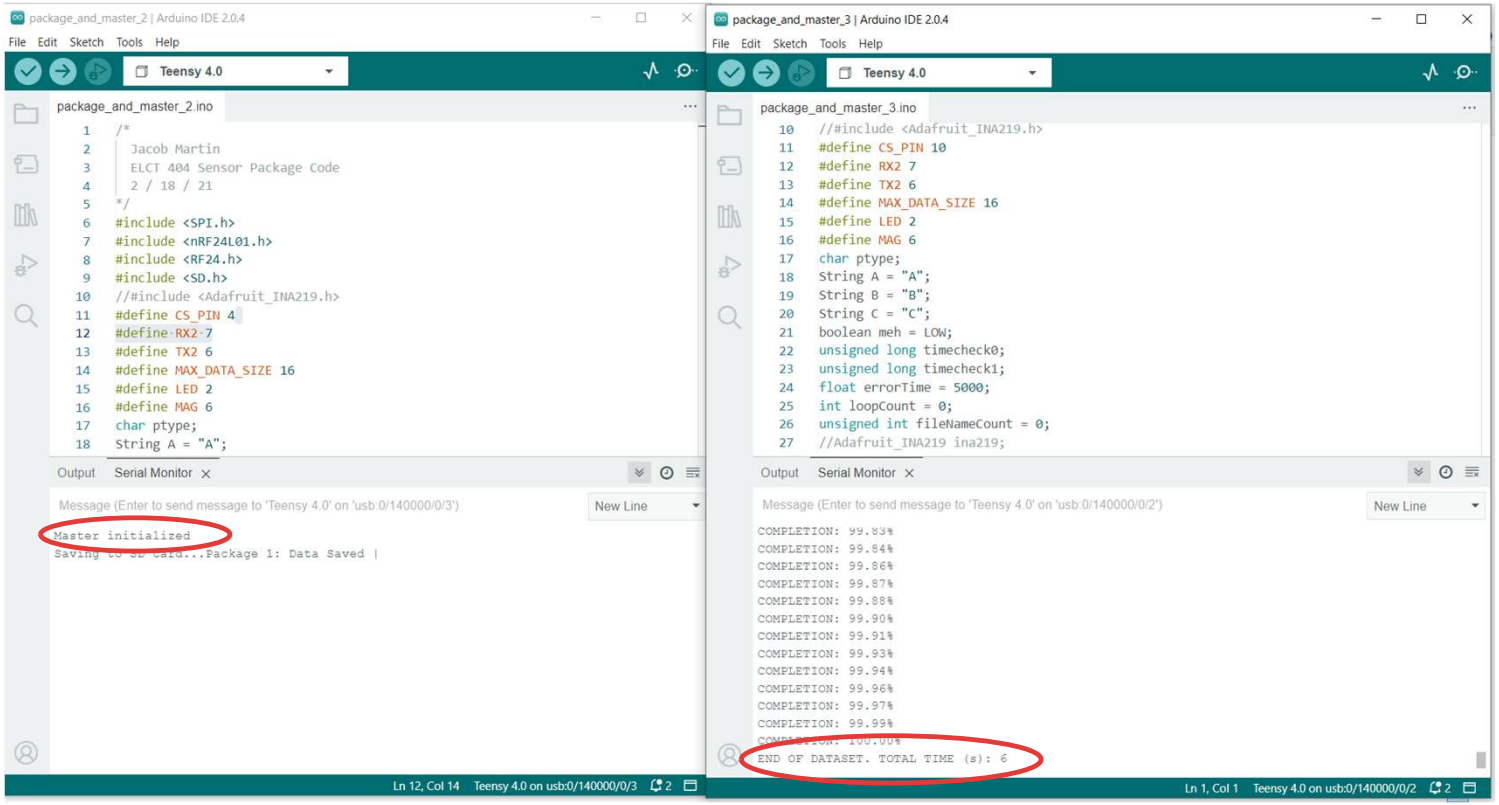
CHIP-TO-CHIP RF OVERVIEW

- The switch from Teensy to Raspberry Pi at the end of last semester complicated chip-to-chip RF significantly
- Three main stages of work
  1. Teensy-to-Teensy Communication
  2. Teensy-to-Pi using Arduino
  3. Teensy-to-Pi using Python

TEENSY-TO-TEENSY COMMUNICATION

- Successfully used Jake Martin's code to transfer a full data packet between two Teensies
  - Served as basis for future csv-transmission tests
  - Helped team get a better grasp of data transfer methods
- This work was presented at McNair Center's Digital Transformation Event
  - DT Event delayed our move from Teensy to Pi due to sensor integration requirements

TEENSY-TO-TEENSY COMMUNICATION



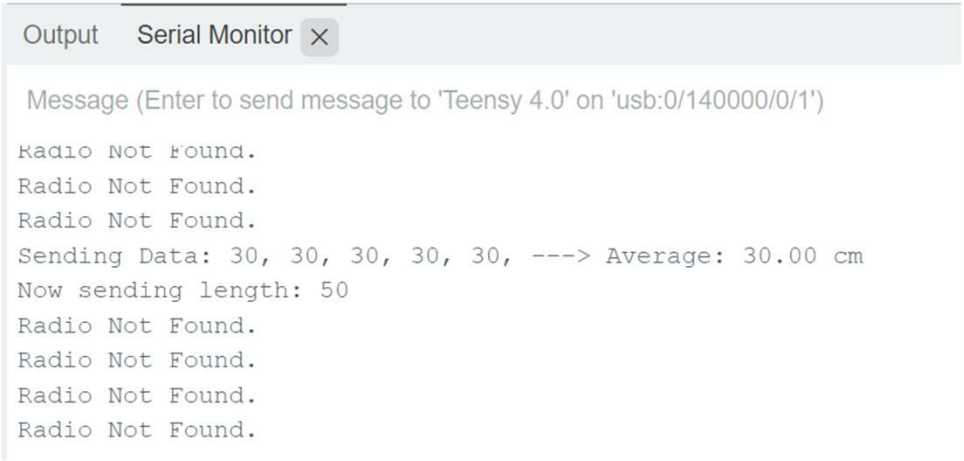
Engineering and Computing

TEENSY-TO-PI USING ARDUINO

- Comprised most of our time on RF comms
- Arduino Create Agent: Cloud-based software that allows Arduino IDE to run on a raspberry Pi
  - Our Plan: Use Create Agent to run Jake's code on the Pi
- We were ultimately unable to compile Arduino code on the Pi
  - Create Agent Initially Unable to connect with Pi
  - Data conversion failures when compiled on Pi
    - "atoi" command used in wireless comms methods failed on Pi

TEENSY-TO-PI USING PYTHON

- Corinne suggested Python
  - Pointed us to the Horus project
- Modified code from Horus Repo used to successfully transfer single floats
  - Further modifications to allow for full csv transmission were not successful
  - Failure seemed to occur when transferring char arrays



Confirmation of Data Transfer from Arduino Transmitter

COMPLICATING FACTORS OF RF COMMS

1. Underestimation of differences between Pi and Teensy
   - Trying to use Arduino on the Pi cost us most of our time
2. Lack of familiarity with relevant programming languages
   - No experience with Python or Arduino
3. Lack of familiarity with RF functionality
   - Buffer and type-casting concerns complicated changes to Horus code

USER INTERFACE

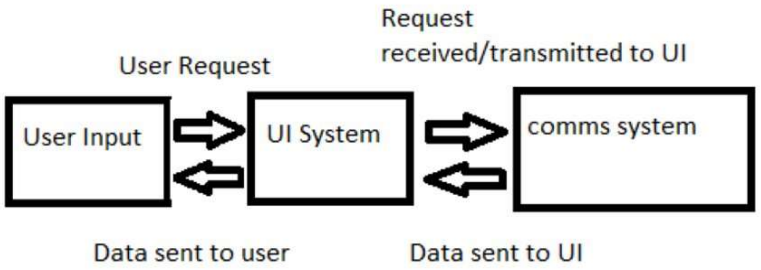
Wyatt Hill

OVERVIEW

- The research and use of a User Interface communication with the Cellular chip significantly complicated data collection and display.
- Four main points
  1. NodeRED pricing for application
  2. NodeRED connectivity implementation
  3. SheetAPI connectivity implementation
  4. UI redundancy

BLOCK DIAGRAM

- UIs investigated: NodeRED and Google Sheets API



PROBLEMS AROUSED

- Need for an IoT service to operate NodeRED script.
- Pricing for nodeRED per month would be $30.
- Accessing the cellular chip not possible due to password protected info, that could not be transmitted through NodeRED.
- NodeRED was then replaced with SheetAPI. Supposed to be able to read text off a website to acquire data.
- Issues arouse with SheetAPI due to inabilty to access cellular chip website's due to password restriction, or lack of data

FUTURE WORK

- Establish communication between cellular communication chip and Raspberry Pi
- Sending CSV file from sensor package to Raspberry Pi
- Accessing processed data through cellular chip interface

APPENDIX


UNIVERSITY OF
South Carolina
College of Engineering
and Computing

CREATE AGENT: RASPBERRY PI SETUP

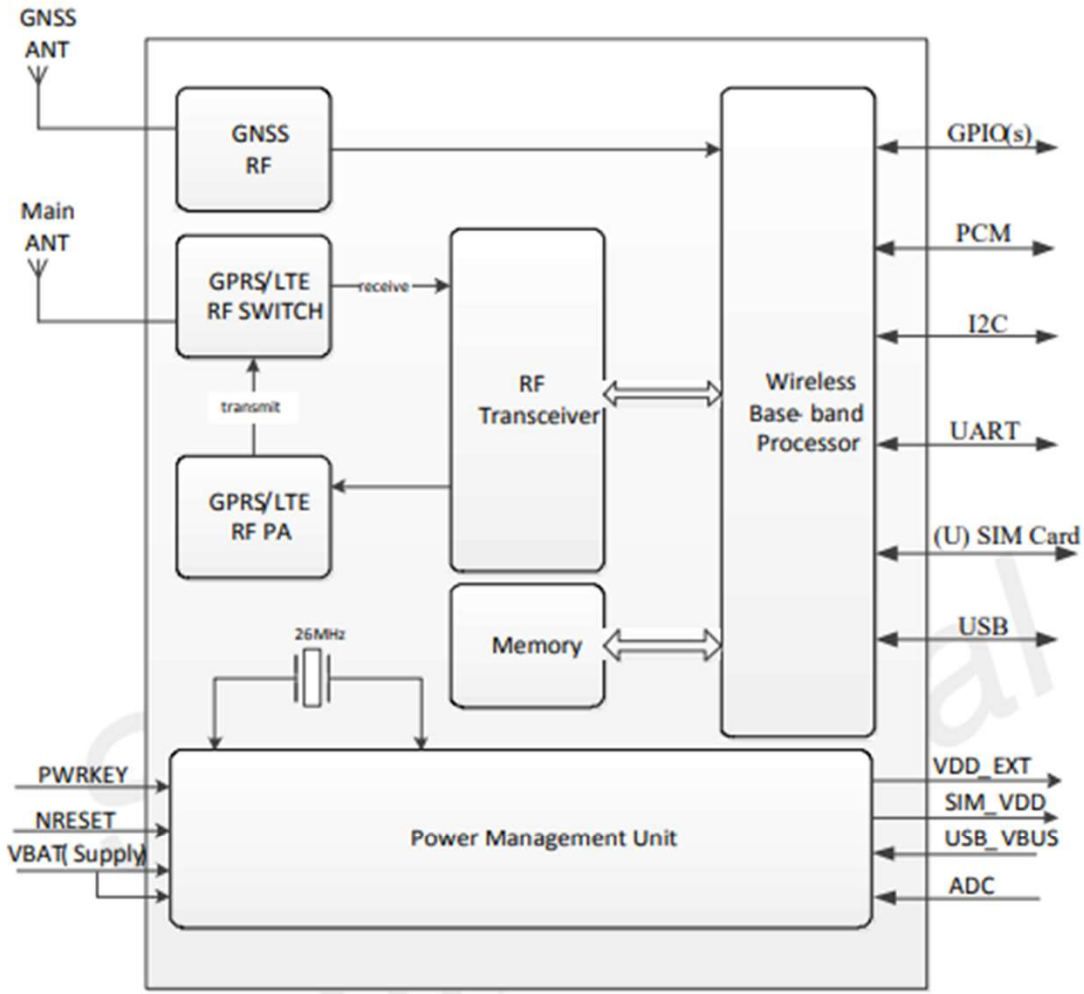
RECEIPT OF RF DATA BY PI

```
kcrean@raspberrypi:~/Downloads/horus-main/RPi_Cron_Job/RF24-master $  /usr/bin/env /bin/python /home/kcrean/.vscode/extensions/ms-
python.python-2023.4.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 53575 -- /home/kcrean/Downloads/horus-main/RP
i_Cron_Job/RF24-master/master.py
No acknowledgment of transmission!
04/18/2023 13:39, 30.00
kcrean@raspberrypi:~/Downloads/horus-main/RPi_Cron_Job/RF24-master $  cd /home/kcrean/Downloads/horus-main/RPi_Cron_Job/RF24-maste
r ; /usr/bin/env /bin/python /home/kcrean/.vscode/extensions/ms-python.python-2023.4.1/pythonFiles/lib/python/debugpy/adapter/../.
./debugpy/launcher 50245 -- /home/kcrean/Downloads/horus-main/RPi_Cron_Job/RF24-master/master.py
04/18/2023 13:40, 30.00
kcrean@raspberrypi:~/Downloads/horus-main/RPi_Cron_Job/RF24-master $
```

CELLULAR CHARACTERIZATION: READ SMS

```
Modem>
+CMTI: "SM",0
R
        ---> AT+CMGF=1
        <--- OK
        ---> AT+CPMS?
        <--- +CPMS: "SM",1,10,"SM",1,10,"SM",1,10

Reading SMS #0
        ---> AT+CMGF=1
        <--- OK
        ---> AT+CSDH=1
        <--- OK
        ---> AT+CMGR=0
+CMGR: "REC UNREAD","+447937405250",,"22/11/28,22:00:32+00",145,36,0,0,"+447797704088",145,46
Test: 28 Nov 2022 from Lab desktop to SIM7000G
***** SMS #0 (46) bytes *****
Test: 28 Nov 2022 from Lab desktop to SIM7000G
*****
Modem>
```



General Features

Global-Band LTE CAT-M1：
B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/B
26/B28/B39;
Global-Band LTE CAT NB-IoT1：
B1/B2/B3/B5/B8/B12/B13/B17/B18/B19/B20/
B26/B28;
GPRS/EDGE 850/900/1800/1900Mhz
Control Via AT Commands

Supply voltage range: 3.0V~ 4.3V, Typ: 3.8V

Operation temperature: -40°C to +85°C

Dimensions: 24 X 24 X 2.6mm

Weight: 3.0g

GNSS (GPS,GLONASS,BeiDou)

Interfaces

USB2.0 x1 (high-speed)

UART x2 (7-wire UART and 2-wire UART

interface multiplex from GPIO)

SIM card x1 (1.8V and 3V)

I2C x1

GPIO x5

ADC x1

PCM x1

Data

LTE CAT-M1(eMTC)

- Uplink up to 375kbps, Downlink up to 300kbps

NB-IoT

- Uplink up to 66kbps, Downlink up to 34kbps

EDGE Class

- Uplink up to 236.8Kbps, Downlink up to 236.8Kbps

GPRS

- Uplink up to 85.6Kbps, Downlink up to 85.6Kbps

Consumption

Power off: 7uA

PSM: 9uA

Sleep: 1mA

Idle: 11mA