BY: OLIVIA KIM,
BAYSIDE HIGH SCHOOL

FROM HARDWARE TO DATA PIPELINE: FABRICATION AND AUTOMATION OF A RASPBERRY PI COSMIC RAY DETECTOR NETWORK.

8/13/26

QUEENSBOROUGH COMMUNITY COLLEGE CUNY

ABSTRACT:

A SOLAR RAY COSMIC DETECTOR IS A DEVICE DESIGNED TO DETECT HIGH ENERGY PARTICLES ORIGINATING FROM THE SUN. THE DETECTED PARTICLES, PARTICULARLY MUONS, ARE PROCESSED THROUGH A DATA ACQUISITION (DAQ) BOX, WHICH COLLECTS, TRANSFERS, AND ANALYZES THE RESULTING DATA. ALTHOUGH THE OVERALL PROCESS MAY APPEAR STRAIGHTFORWARD, DEVELOPING AND MAINTAINING THE DAQ BOX REQUIRES CAREFUL ATTENTION TO DETAIL, AS AN ERROR IN ANY COMPONENT CAN AFFECT THE FUNCTIONALITY OF THE ENTIRE SYSTEM. TROUBLESHOOTING THEREFORE INVOLVES SYSTEMATICALLY EXAMINING THE CODE, WIRING, CIRCUIT BOARDS, AND OTHER HARDWARE COMPONENTS TO IDENTIFY AND CORRECT POTENTIAL SOURCES OF ERROR. AS PART OF THIS RESEARCH, I CONTRIBUTED TO THE DEVELOPMENT OF AN AUTOMATED DATA TRANSFER SYSTEM. SPECIFICALLY, I HELPED WRITE A PROGRAM THAT RETRIEVES DATA COLLECTED BY A RASPBERRY PI, A COMPUTER INTEGRATED INTO THE DAQ SYSTEM, AND AUTOMATICALLY TRANSFERS THE DATA TO A GOOGLE DRIVE(OR ANY CLOUD). THIS AUTOMATION ALLOWS THE RASPBERRY PI TO COLLECT AND TRANSMIT DATA REMOTELY, ENABLING RESEARCHERS TO ACCESS THE MEASUREMENTS REGARDLESS OF THE DEVICE'S PHYSICAL LOCATION. BY STREAMLINING THE DATA-TRANSFER PROCESS, THE SYSTEM IMPROVES ACCESSIBILITY, AND EFFICIENCY IN DATA COLLECTION AND ANALYSIS. IN ADDITION TO THE SOFTWARE DEVELOPMENT COMPONENT, I CONTRIBUTED TO THE CONSTRUCTION OF A CIRCUIT BOARD FOR THE DAQ BOX. THIS WORK PROVIDED EXPERIENCE IN INTEGRATING HARDWARE AND SOFTWARE COMPONENTS AND DEMONSTRATED THE IMPORTANCE OF PRECISION AND RELIABILITY WHEN DEVELOPING SCIENTIFIC INSTRUMENTATION.

INTRODUCTION:

AUTOMATING DATA TRANSFER TO GOOGLE DRIVE REQUIRES AN UNDERSTANDING OF THE RASPBERRY PI'S FUNCTIONALITY, ITS ROLE WITHIN THE DAQ BOX, AND THE EXISTING CODE DEVELOPED BY PREVIOUS RESEARCHERS FROM OUR PROGRAM. AN ARDUINO BASED SYSTEM WAS PREVIOUSLY PROGRAMMED TO TRANSFER COLLECTED DATA TO THE RASPBERRY PI. BUILDING ON THIS FOUNDATION, MY CO-MENTOR AVAKASH AND I MODIFIED A GRAPHICAL USER INTERFACE (GUI) DEVELOPED BY OUR MENTOR, PROFESSOR KOFI, TO IMPROVE ITS ORGANIZATION, READABILITY, AND EFFICIENCY. WE IMPLEMENTED BIDIRECTIONAL FILE TRANSFER BETWEEN THE RASPBERRY PI AND THE CENTRAL COMPUTER. AS WELL AS AN AUTOMATED FILE-TRANSFER FEATURE USING RCLONE. THESE FUNCTIONS ARE CURRENTLY ACCESSIBLE THROUGH THE MODIFIED GUI. IN ADDITION, I GAINED HANDS-ON EXPERIENCE WITH PCB DEVELOPMENT, INCLUDING CIRCUIT-BOARD ASSEMBLY, SOLDERING, AND DESIGNING CUSTOM PCBS USING FUSION. WHILE MY PROJECT PARTNERS FOCUSED MORE EXTENSIVELY ON THE HARDWARE DEVELOPMENT, THIS WORK PROVIDED ME WITH VALUABLE EXPERIENCE INTEGRATING SOFTWARE AND HARDWARE WITHIN THE DAQ BOX.

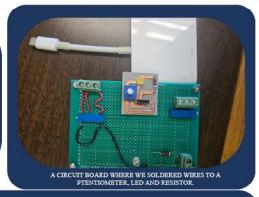
METHODS AND MATERIALS:

RCLONE

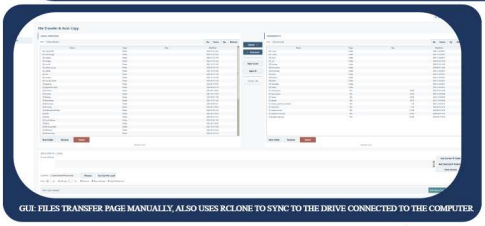
RCLONE, A COMMAND-LINE TOOL FOR MANAGING AND TRANSFERRING FILES, WAS USED TO AUTOMATE DATA TRANSFER BETWEEN THE RASPBERRY PI AND THE CENTRAL COMPUTER. INITIAL ATTEMPTS TO IMPLEMENT THE TRANSFER WITHOUT RCLONE WERE COMPLICATED BY SOFTWARE-INSTALLATION RESTRICTIONS ON THE RASPBERRY PI. WE ALSO EXPLORED USING PUTTY TO ESTABLISH A CONNECTION BETWEEN THE RASPBERRY PI AND WINDOWS COMPUTER; HOWEVER, THE DEVICES WERE CONNECTED TO DIFFERENT NETWORKS, PREVENTING THE REQUIRED NETWORK CONFIGURATION. FOR THE HARDWARE COMPONENT, WE ARE DEVELOPING CUSTOM PCBS USING A BAMBU LAB H2D 3D PRINTER. THIS PROCESS HAS PRESENTED CHALLENGES IN ACHIEVING PRECISE DIMENSIONS, PARTICULARLY FOR SMALL PCB HOLES, WHICH THE PRINTER MAY PARTIALLY OR COMPLETELY FILL DUE TO THEIR SIZE. THESE ISSUES HAVE REQUIRED CONTINUED DESIGN ADJUSTMENTS AND REFINEMENT OF THE FABRICATION PROCESS.

RESULTS:

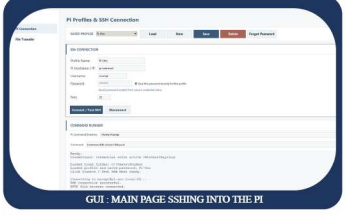
OUR PROGRESS HAS BEEN SUCCESSFUL, ALTHOUGH THE SYSTEM IS NOT YET FULLY AUTOMATED. WE DEVELOPED A GUI CAPABLE OF MANUALLY TRANSFERRING DATA FROM THE RASPBERRY PI TO GOOGLE DRIVE AND TESTED RCLONE-BASED FILE TRANSFERS THROUGH THE TERMINAL. HOWEVER, THE RASPBERRY PI'S LIMITED NETWORK CONNECTIVITY HAS CREATED SIGNIFICANT CHALLENGES. BECAUSE THE PI LACKS A DIRECT INTERNET CONNECTION, IT RELIES ON THE GUI'S CONNECTION DURING DATA TRANSFERS, LIMITING OUR ABILITY TO DIRECTLY ACCESS AND MANAGE ITS FILES. TO ADDRESS THIS ISSUE, WE ARE TESTING A HOTSPOT-BASED CONNECTION TO PROVIDE THE RASPBERRY PI WITH INDEPENDENT INTERNET ACCESS AND IMPROVE THE RELIABILITY OF DATA TRANSFER.



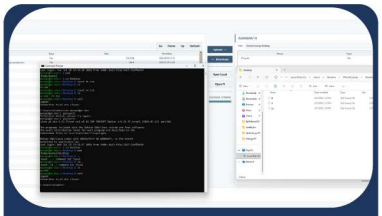
A CIRCUIT BOARD WHERE WE SOLDERED WIRES TO A PTENTIOMETER, LED AND RESISTOR.



GUI: FILES TRANSFER PAGE MANUALLY, ALSO USES RCLONE TO SYNC TO THE DRIVE CONNECTED TO THE COMPUTER



GUI : MAIN PAGE SSHING INTO THE PI



CHECKING FILE TRANSFER MANUALLY(VIA TERMINAL)

```
88  import subprocess
89  import sys
90  import time
91  import json
92  from pathlib import Path
93  from datetime import datetime, timezone
```

IMPORTING LIBRARIES IN ORDER TO CODE THE AUTOMATION OF DATA TRANSFER

CONCLUSION:

ALTHOUGH WE HAVE SUCCESSFULLY DEVELOPED A GUI FOR DATA TRANSFER, THE SYSTEM STILL RELIES ON THE GUI FOR OPERATION. TO ACHIEVE A FULLY AUTOMATED AND INDEPENDENT SOLUTION, WE ARE IMPLEMENTING RCLONE WITH CRON JOBS TO AUTOMATE THE TRANSFER OF COLLECTED DATA TO CLOUD STORAGE. THIS APPROACH WILL ALLOW DATA TO BE CONTINUOUSLY COLLECTED AND UPLOADED WITHOUT REQUIRING THE GUI, MAKING IT REMOTELY ACCESSIBLE FROM VIRTUALLY ANYWHERE AND IMPROVING THE OVERALL RELIABILITY AND EFFICIENCY OF THE SYSTEM.

CREDITS:

THIS WORK WAS SUPPORTED BY THE CUNY STEM RESEARCH ACADEMY THROUGH A GRANT FROM THE PINKERTON FOUNDATION. I ALSO WANT TO THANK JAMES, NOUR, PROFESSOR ARMENDARIZ, PROFESOR KOFI, CO-MENTOR AVAKASH, CO-MENTOR SHORN AND THE RESEARCH PHYSICS DEPARTMENT.

```
"""
Runs a couple of cheap rclone commands to fail fast with a clear
error message, instead of a confusing crash halfway through a
transfer.
"""
try:
    subprocess.run(
        ["rclone", "version"],
        capture_output=True, check=True, text=True
    )
except FileNotFoundError:
    log.error("rclone is not installed. Run: sudo apt install rclone")
    return False
except subprocess.CalledProcessError as e:
    log.error("rclone --version failed: %s", e.stderr)
    return False

# `rclone lsd <remote>:` lists top-level folders on the remote.
# If the remote name is wrong or not authenticated, this fails.
result = subprocess.run(
    ["rclone", "lsf", f"{REMOTE_NAME}:"],
    capture_output=True, text=True
)
if result.returncode != 0:
    log.error(
        "Cannot reach remote '%s'. Did you run `rclone config`? Details: %s",
        REMOTE_NAME, result.stderr.strip()
    )
    return False
```

SNIPPET CHECKS WHETHER THE SYSTEM OR THE PI HAS RCLONE