Calvin Trujillo

SID: 862196581

EMAIL: ctruj011@ucr.edu

Custom Lab Report – CS120B

Introduction:

Security is an important part of our current world. Most of the items we interact with day to day have some sort of security protocol implemented within it. Our phones have systems that encrypt our messages being sent to other individuals. We are allowed to create passwords for the important pieces of information we want to keep away from prying eyes.

The program will create a randomly generated password through some sort of algorithm and will display it to the user. The generated code will take the form of a 4-digit number. This code can then be used to access some sort of data, in this case a file.

*So far, the only item I have had trouble with implementing is the LCD due to unforeseen situations. (Mostly dud displays) The system was going to have an IR sensor to remotely turn off and, on the system, but I have decided that it would not make sense to include this system. I thought it would add another "vulnerability" to the system. Another issue that was run into was the memory allocation was not enough to fully run all the devices at once. I needed to change the amount of data being used to allow all devices to run correctly. Was not able to completely display the code to the User using the screen; however, the serial monitor was a bit different.

Complexities:

Complexities created for the project are:

1. RNG – Creation of the RNG library which works similarly to calling the function Rand(), but will be based on the Park Miller model. This model no longer requires the system to set a certain seed and will continue to randomize the code as the system continues. It considers the amount of time that the system has been running into its calculation. The library will not allow repeat numbers and has parameters in place.
    a. Parameters are as follows:
        i. No numbers will repeat in a sequence of 4 (i.e., 4444)
        ii. No numbers will repeat again (i.e., 2354, 2354 being output twice)
    b. Imitations:
        i. The code can only generate 100 different amounts of numbers.
            1. Need to work around the Arduino memory allocation.
            2. Cannot use dynamic memory to release the extra data.

2. GY-521 module – it's a module which provides the user with the temperature, acceleration in a 3D axis and Gyro Scope functionality.
    a. Module is an I2C device that communicates with the R3 UNO Board.

   b. The state machine responsible for running this device will sample the device's information every second.
      i. This sample is supposed to simulate someone "removing" the panel and attempting to break through the security.
      ii. Once movement is detected and determined to be tinkering, it will send a single to brick the entire system. (Not implemented is the way to fix it due to time constraints.)

3. Nokia or TFT LCD Library – a LCD library which shows the user information such as the code and a message saying the system is being bricked.
   a. Creation of the library for TFT was not successful, if TFT is being used, then it's the library created by ADAFRUIT. Only being used in the case that the NOKIA 5110 screens are not functional.
   b. Both library "Implementations Attempts" are provided with the source code for this project.

User Guide:

The user will take their FID card or chip and place it onto the RFID scanner. The scanner will send a message to both the Serial Monitor and LCD display with the message that states whether the ID has access granted or not. Then a code will be generated and displayed to the user. This code can then be used in either everyday life or to access a certain file implemented within the system. (This file will not be accessed during the demo of the video as I planned on the user to take the code to a separate system and implement the code.)

If the user so far as chooses to attempt to tinker with the device in any way, shape, or form, the system will entirely brick itself. Through the attempt of a physical tinker, the module responsible for sending the gyro scope information will trigger an allocated flag which will then brick the system by outputting a message.

The serial monitor will be on display for the user to show them the information. (In the case of the LCDs not being implemented correctly, we will show the actual information on the Serial monitor.)

Hardware Components:

Hardware components being used in this project are:

1. R3 UNO Board
    a. The main controller which allows all the other modules to interact.
    b. Arduino is the main program which allows the code to be uploaded to this controller.
2. Breadboard
    a. Allows for the physical connections between the power of the board and the components.
    b. Also allows wires to be placed and allows us to communicate with the main controller.
3. Jumper Wires
    a. Allow for communication and data transfer to occur
4. LCD (Nokia 5110 84x48 or TFT 1.44 128x128)
    a. The main LCDs being used in the project
    b. Implementation was difficult and possibly requires more attention.
5. GY-521 (Gyro Scope Portion)
    a. Movement system which will set off a trigger for the execution of the entire system.
    b. If the system is moved physically, then the entire system becomes bricked.
        i. Simulates an individual attempting to break into the system through the removal of the LCD.
6. RFID Module (Provided by R3 Starter Kit)
    a. Sensor which checks the ID found on either the card or chip

Software:

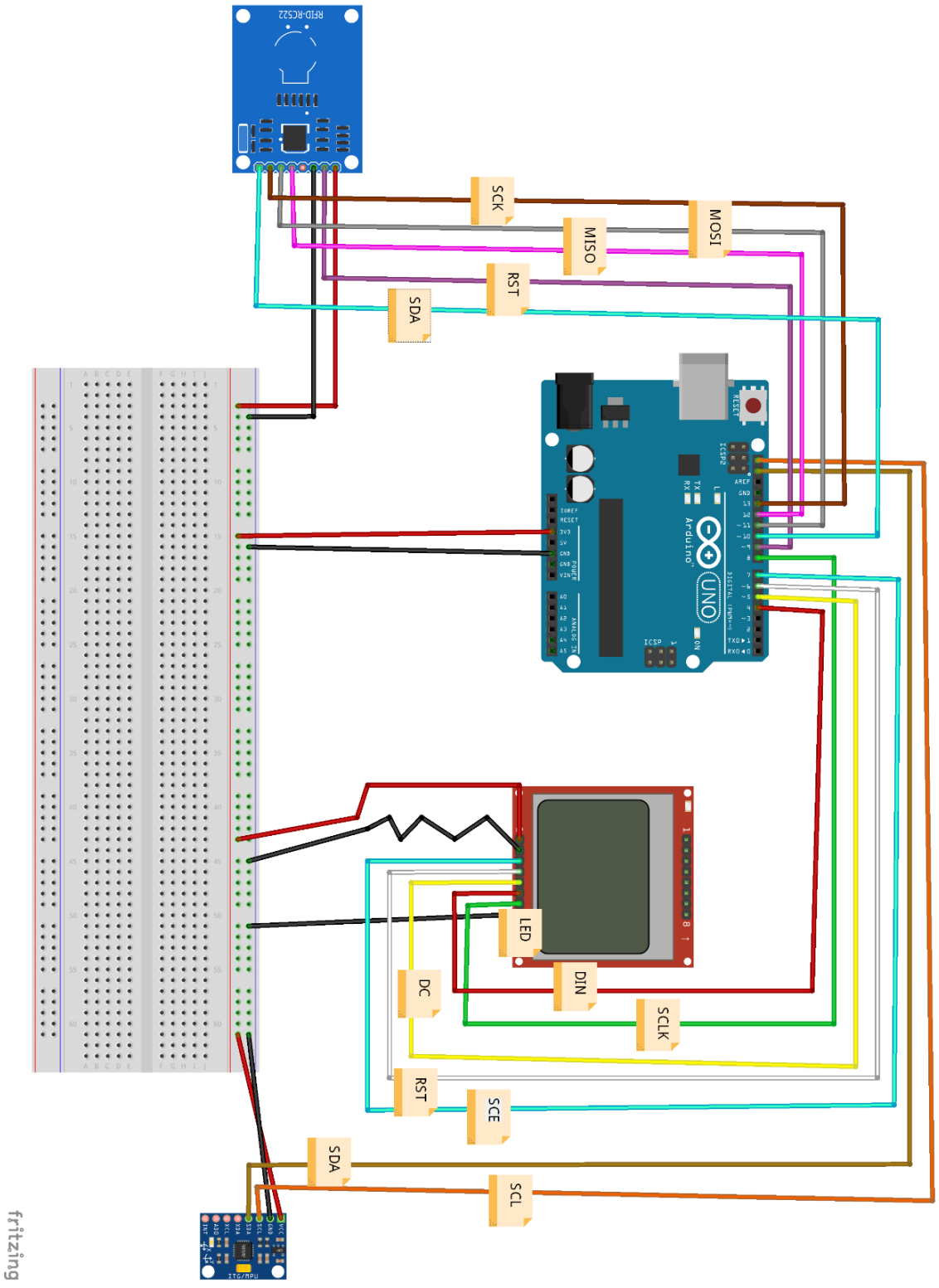
Software libraries being used in this project are:

1. SPI
    a. Attempted implementation of the TFT LCD through SPI library.
    b. A note will be provided if implementation was successful.
2. Wire
    a. Implementation of the I2C device.
    b. Wire functions provided information for possible bricking.
3. Nokia (Created by Me) //if screen remotely works with all other libraries
    a. Nokia Library Complexity, Data Sheet was useful in AIDING creation
4. TFT ST7735 (Created by ADAFRUIT) //if the Nokia Screen is not functioning
    a. INCASE SPI IMPLEMENTATION WAS A BUST
    b. The video will have a comment about the implementation.
5. String Library
    a. C++ string library to be able to run strings
6. Vector Library
    a. Creating vectors because they look better (I also thought they were able to do dynamic memory management, but Arduino does not allow for such functionality)

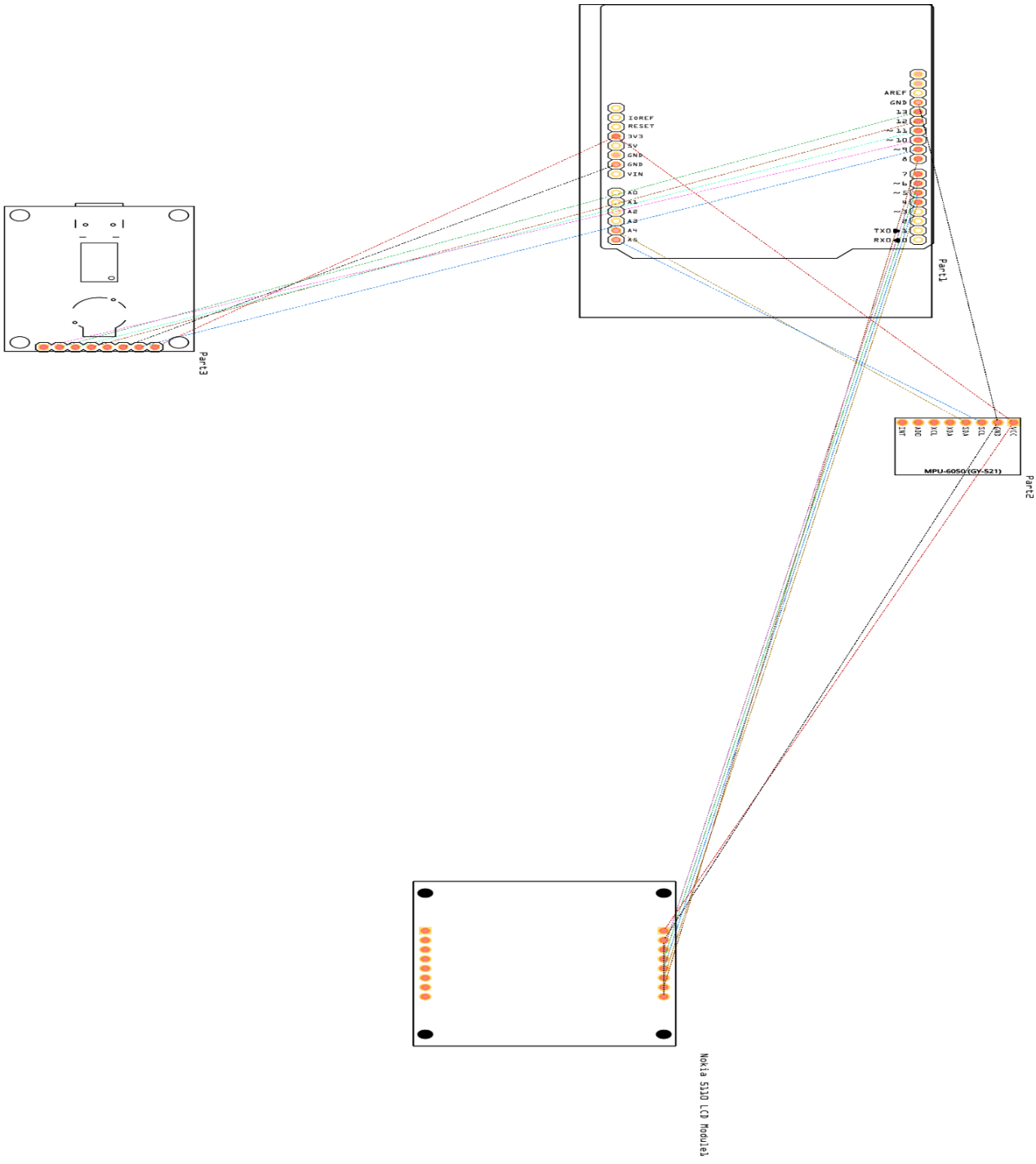
7. STD.IO library
    a. Don't remember why this was implemented, but code runs with it commented out.
8. RNG Library (Created by Me with the information provided by Parker Miller PDF)
    a. Parker Miller Research Paper was provided by the Tas and Professor.
    b. The RNG library uses the seed given by the research paper and adds a timer to allocate a randomized movement. Since the system will not be calling the RNG system repeatedly.
        i. Time allotted by the function millis() will do this for us.

Wire Diagrams/Schematics/Circuit Design: Assuming Nokia LCD being used

-- Circuit Diagram using Breadboards --

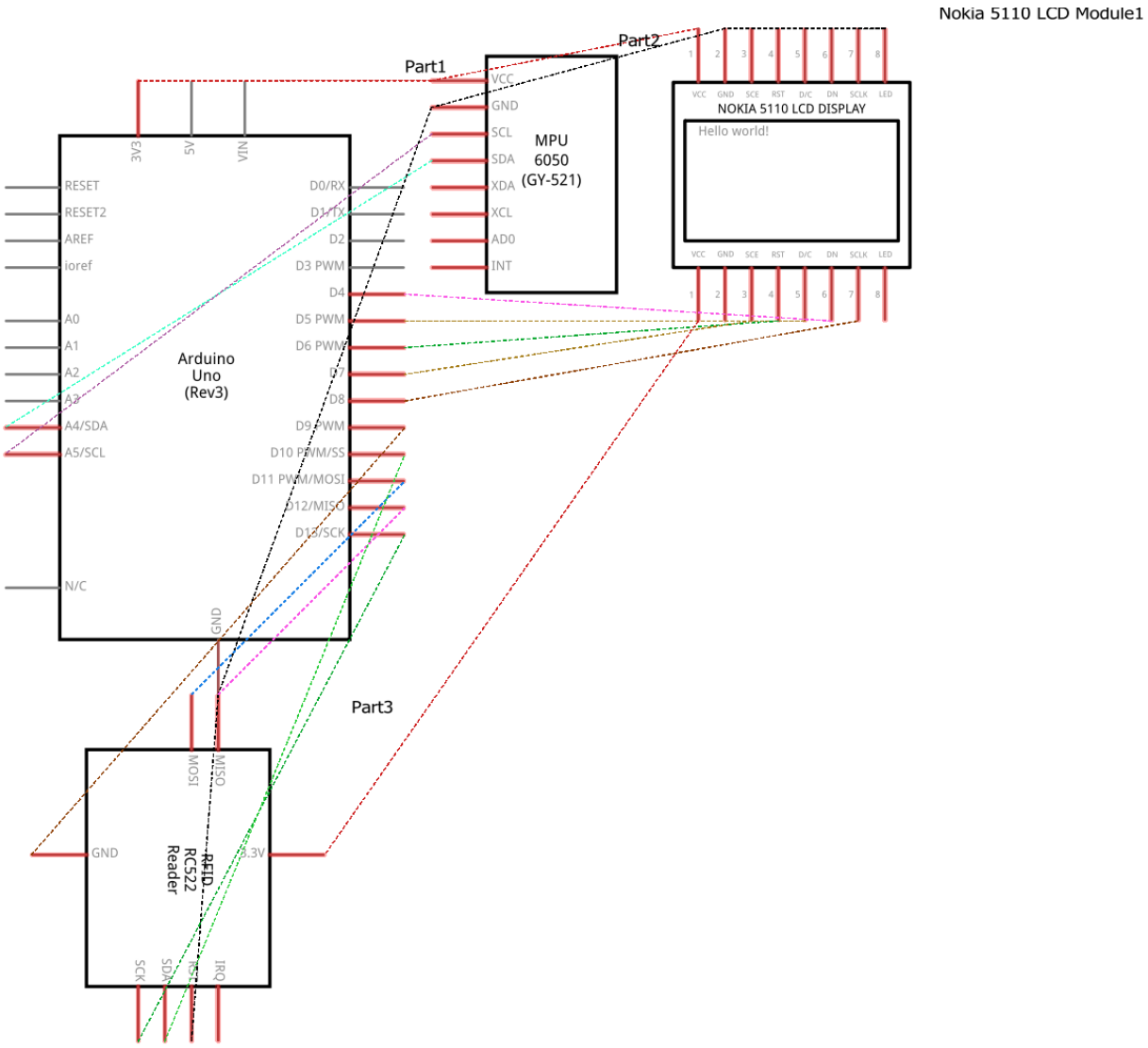


-- Schematics of the Board --

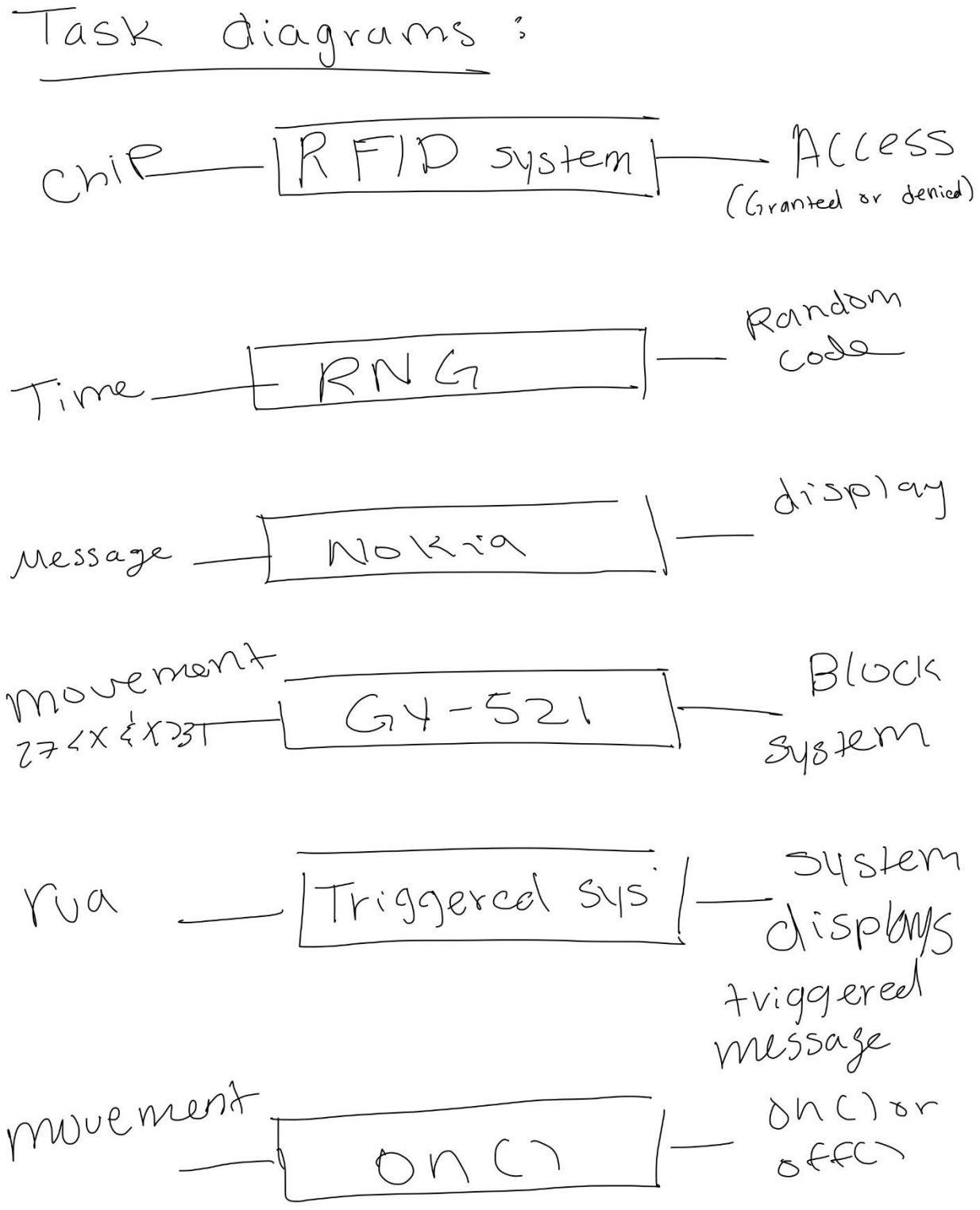


fritzing

-- PCB Design Diagram --



-- Task Diagrams --

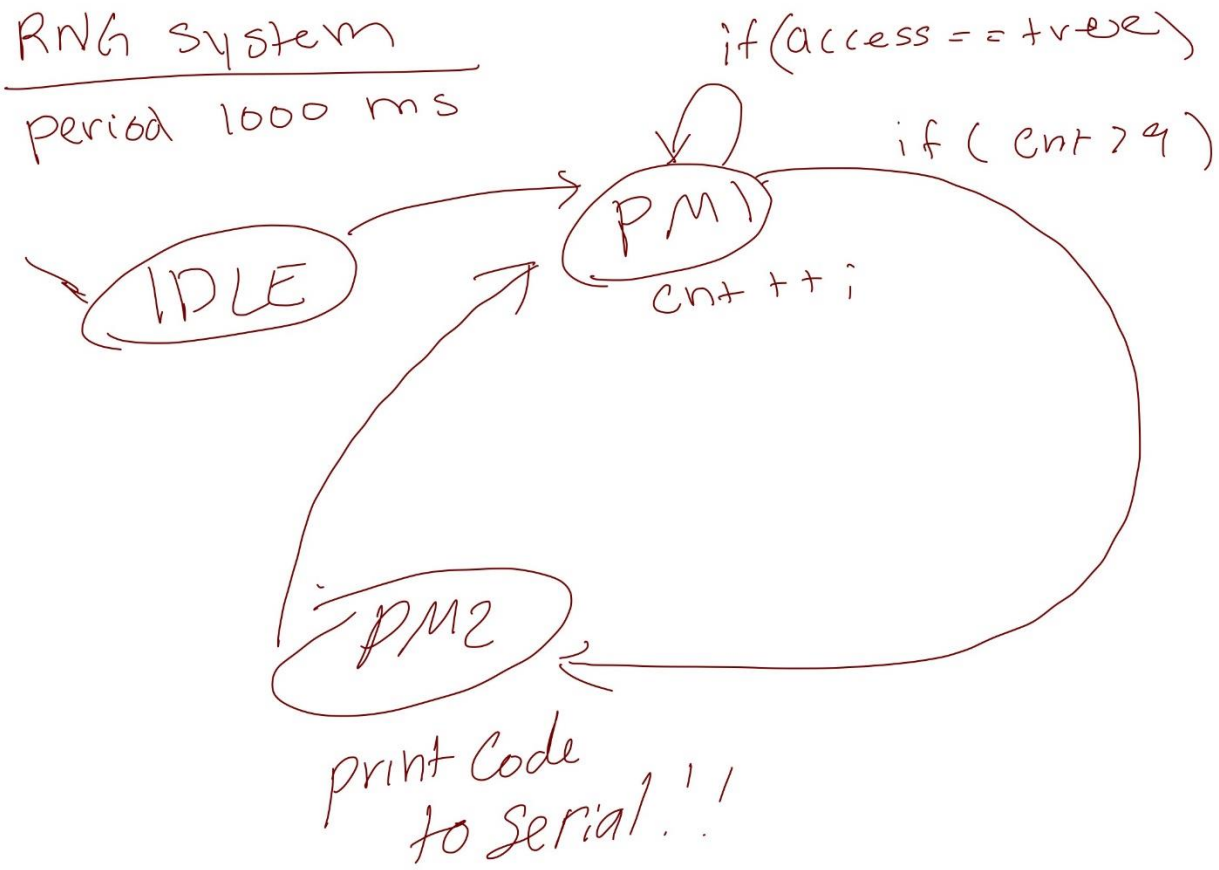


-- SynchSM Diagrams --

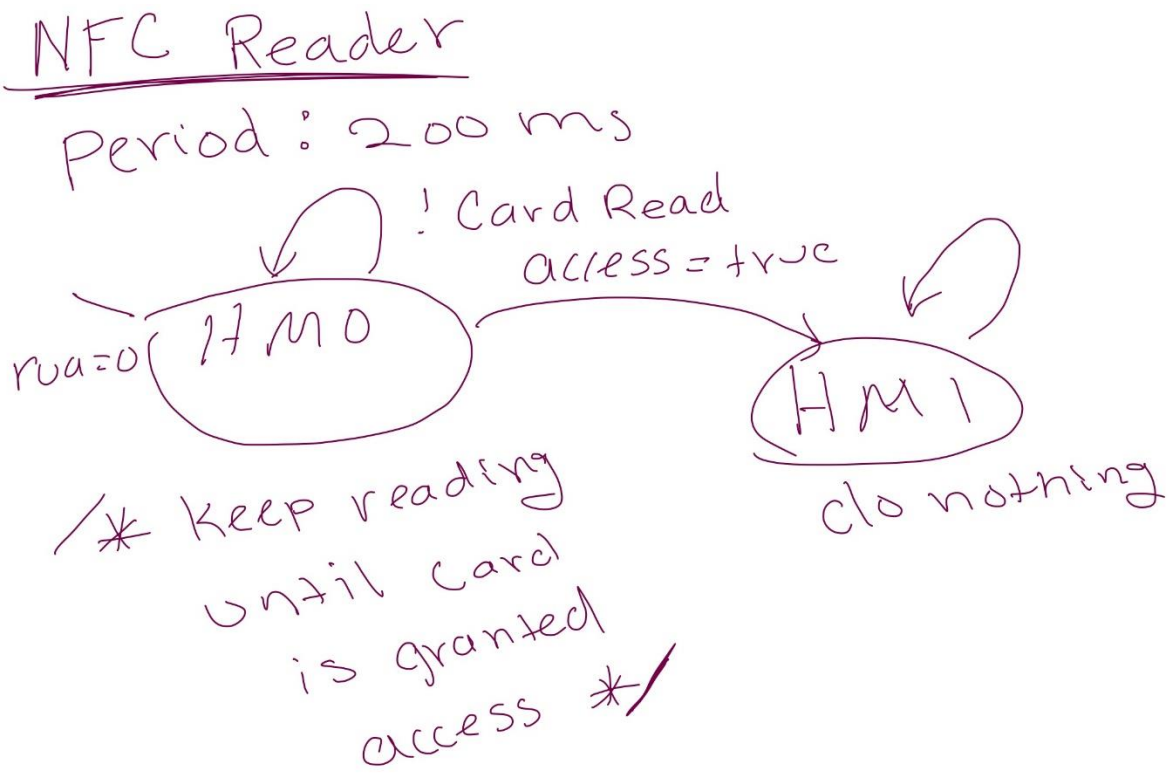
Synch SMs

Global Variables:

```
int once = 0;
int cn = 0;
unsigned in p = 0;
unsigned char x;
int X, Y, Z;
unsigned int ua = 0;
bool access = false;
bool on() {
    bool jim = false;
    if (Y > 28 && Y < 232) {
        jim = true; }
    else {cnt++; }
    return jim; }
Unsigned int Cnt;
```

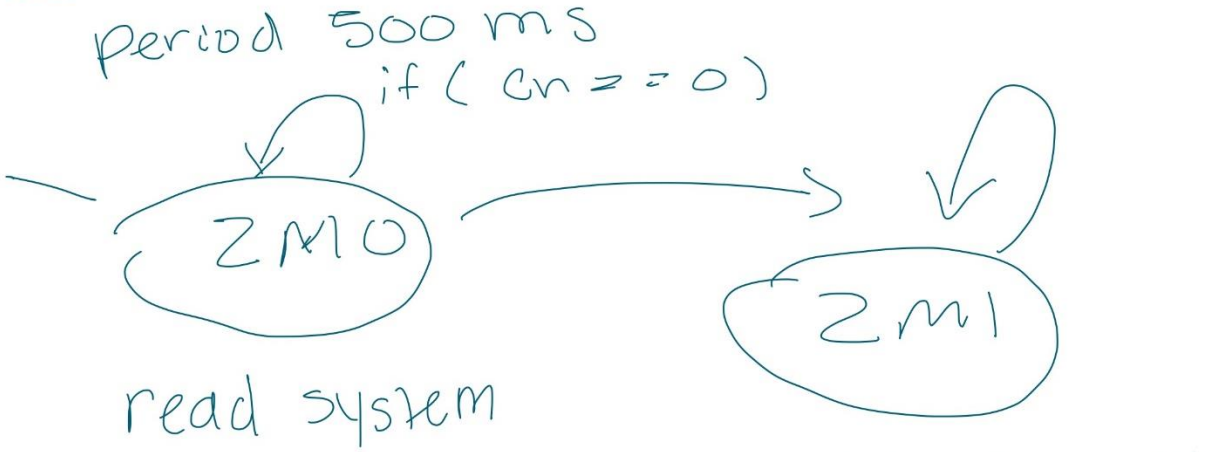


RNG System needs to run different functions inorder to work. Needs to run in Conjunction with CRY521.



NFC Reader lets us access the actual system and allows All the functions to work.

Gyro-521 Module Read



read system

Wire.read(); Printf("Tinkering!")

Gy-521 will block all the system's functions if movement is detected. System breaks!!