# Overview

This is a private repository for my project at WISEST lab during Summer 2018 as an undergraduate research assistant. This project is a dairy IoT, using chiefly MATLAB and Arduino (C++). MATLAB is responsible for gathering data from experiments and finding calibration coefficients. Arduino programs a microcontroller board. The goal is to monitor the core temperature of livestock and to wirelessly send the data through the internet.

# Requirements

1. Arduino Uno connected to your PC via cable
2. The RFID and sensor circuit
3. The sensor
4. MATLAB 2018a (earlier versions might not support all functionalities) installed
5. MATLAB Support Package for Arduino Hardware installed and setup properly

# What these programs do

Run\_Improved\_V2.mlapp gathers the number of samples and displays the raw data plot and the calibrated plot. Linear regression coefficients can be applied and adjusted before the reading starts. After the reading has done, the data will be save automatically and once again before the program is closed. The LED light indicates the data availability. It becomes yellow when there is a problem with the input data, becomes red if no input data is read, turns black when the program finishes or is ready. Once data is collected, it is automatically saved to a CSV file with a timestamp.

Run\_Monitoring.mlapp is similar to Run\_Improved\_v2.mlapp. It keeps reading data until it is stopped by users. It has an ability to breakdown data and save data into multiple files every X second specified by users. In addition to those LEDs, Run\_Monitoring.mlapp has a blue LED indicating a pause. Unlike the first program, this program does NOT save when closed the program window. However, the data should be saved when “Stop” is pressed.

# How to use "Run\_Improved\_V2.mlapp" for known sample number

1. Connect Arduino to your PC.
2. Double-click on the program “Run\_Improved\_V2.mlapp”.
3. Change port name to match the port Arduino connected to. You can also check with Device Manager in your Control Panel and look for Port section.
4. Click “Connect” button to open serial communication between your PC and Arduino.
5. Adjust the number of samples you wish.
6. Check a box called “Activate” and adjust the coefficients properly, if needed.
7. Place RFID antenna close to the sensor. The LED on the RFID reader should blink, indicating that there is a data.
8. Click “Start” button when you are ready.
9. Wait until the LED turns black again.
10. If you wish to collect more data, simply click “Reset” again and press “Start”
11. Repeat step 8-10.
12. Once you are satisfied with data, close the window. Doing this prompts a dialogue block.
13. Select “Yes” if you wish to exit and save data or select “No” if you accidentally click on the close button.

# How to use "Run\_Monitoring.mlapp" for monitoring

1. Connect Arduino to your PC.
2. Double-click on the program “Run\_Monitoring.mlapp”.
3. Change port name to match the port Arduino connected to. You can also check with Device Manager in your Control Panel and look for Port section.
4. Click “Connect” button to open serial communication between your PC and Arduino.
5. Check a box for “save multiple files” if wish to divide data into many smaller files and specify how long in second between each file.
6. Check a box for “activate” and adjust the coefficients properly, if require calibrated data.
7. Place RFID antenna close to the sensor. The LED on the RFID reader should blink, indicating that there is data.
8. Click “Start” button when you are ready.
9. If necessary, click pause to temporarily stop the program. Note that this does not stop the timer within the program.
10. When achieve enough data as desired, click stop to stop reading and save the data.
11. If wish to collect more data, press “Start”.
12. Repeat step 8-11.
13. Once you are satisfied with data, close the window. Doing this prompts a dialogue block.
14. Select “Yes” if wish to exit or select “No” if accidentally click on the close button.

# Troubleshooting