Sensor Data Embedding Device – Purdue Internship

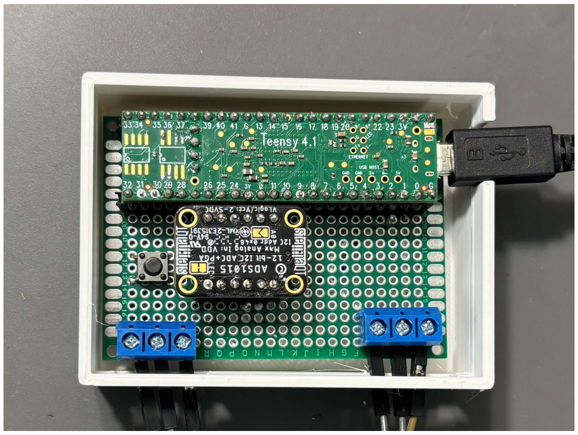
Project Overview

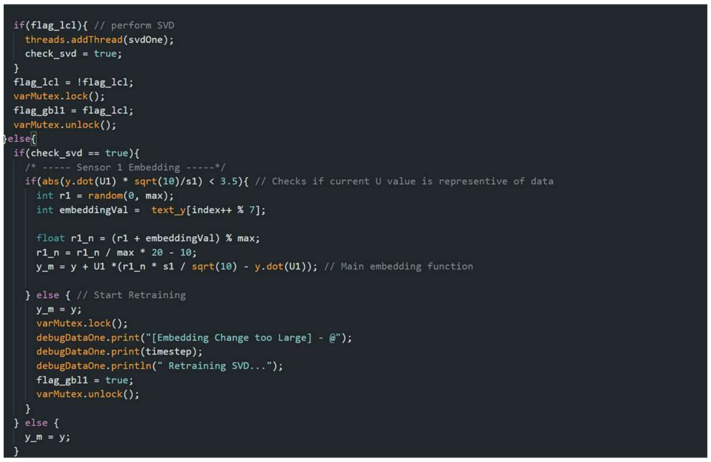
This project implemented a algorithm developed by a postdoctoral researcher to embed text and numerical information directly into sensor data streams. The system enables recovery of embedded information without compromising the integrity of the original sensor data, supporting advanced data verification and tracking use cases.

Technical Details

The system was built around a Teensy 4.1 microcontroller and an ADS115 analog-to-digital converter. Firmware was written in C++ and deployed on a custom perfboard-based circuit. A 3D-printed enclosure was designed for durability and organization. A secondary data generator device was developed to validate performance using over 10,000 real-world data points, with MATLAB scripts used to recover and verify the embedded information.

Images





```
  if(flag_lcl){ // perform SVD
    threads.addThread(svdOne);
    check_svd = true;
  }
  flag_lcl = !flag_lcl;
  varMutex.lock();
  flag_gbl1 = flag_lcl;
  varMutex.unlock();
}else{
  if(check_svd == true){
    /* ----- Sensor 1 Embedding -----*/
    if(abs(y.dot(U1) * sqrt(10)/s1) < 3.5){ // Checks if current U value is representive of data
      int r1 = random(0, max);
      int embeddingVal =  text_y[index++ % 7];

      float r1_n = (r1 + embeddingVal) % max;
      r1_n = r1_n / max * 20 - 10;
      y_m = y + U1 *(r1_n * s1 / sqrt(10) - y.dot(U1)); // Main embedding function

    } else { // Start Retraining
      y_m = y;
      varMutex.lock();
      debugDataOne.print("[Embedding Change too Large] - @");
      debugDataOne.print(timestep);
      debugDataOne.println(" Retraining SVD...");
      flag_gbl1 = true;
      varMutex.unlock();
    }
  } else {
    y_m = y;
  }
```