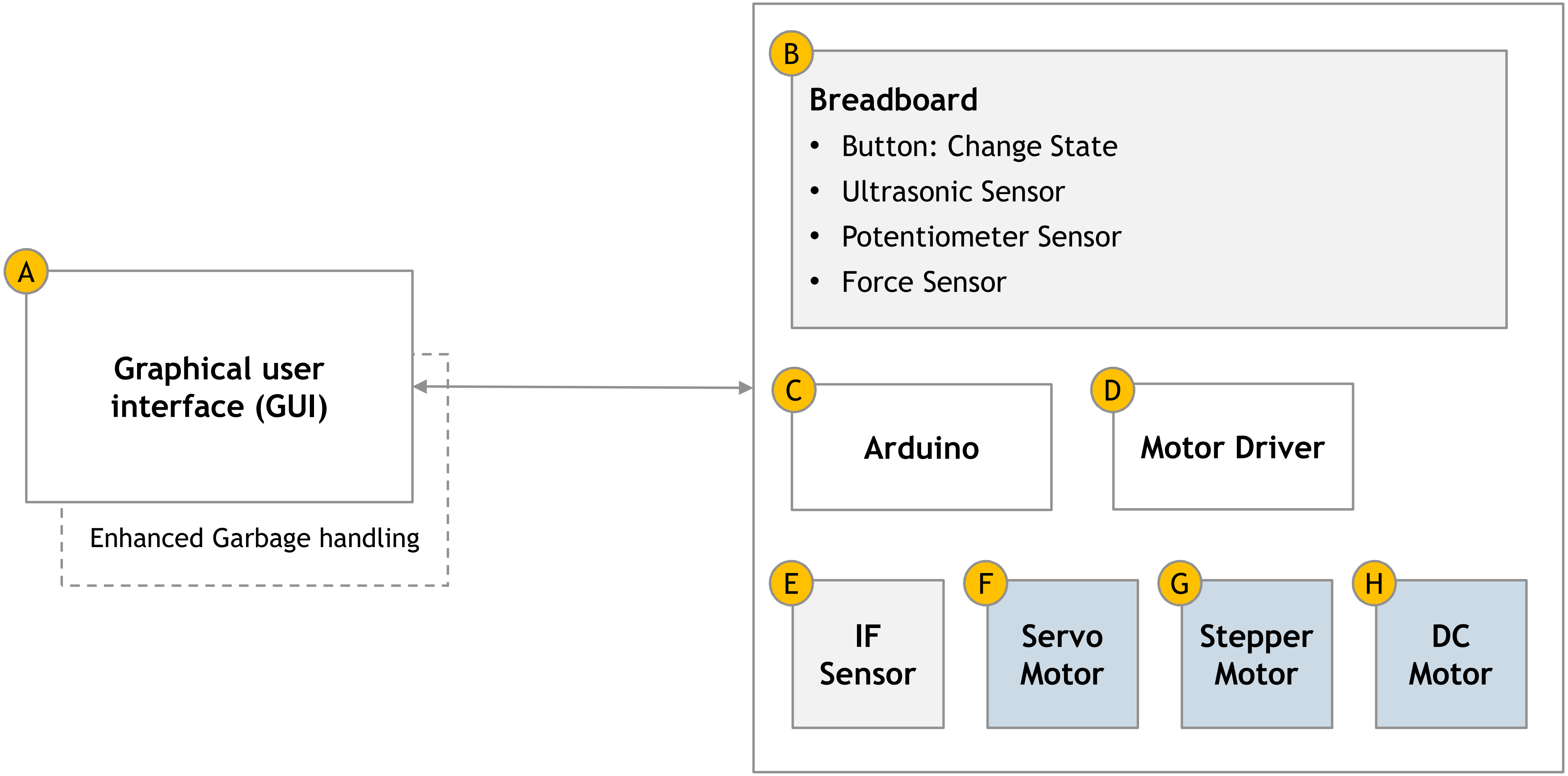
# FlySense

## Team C

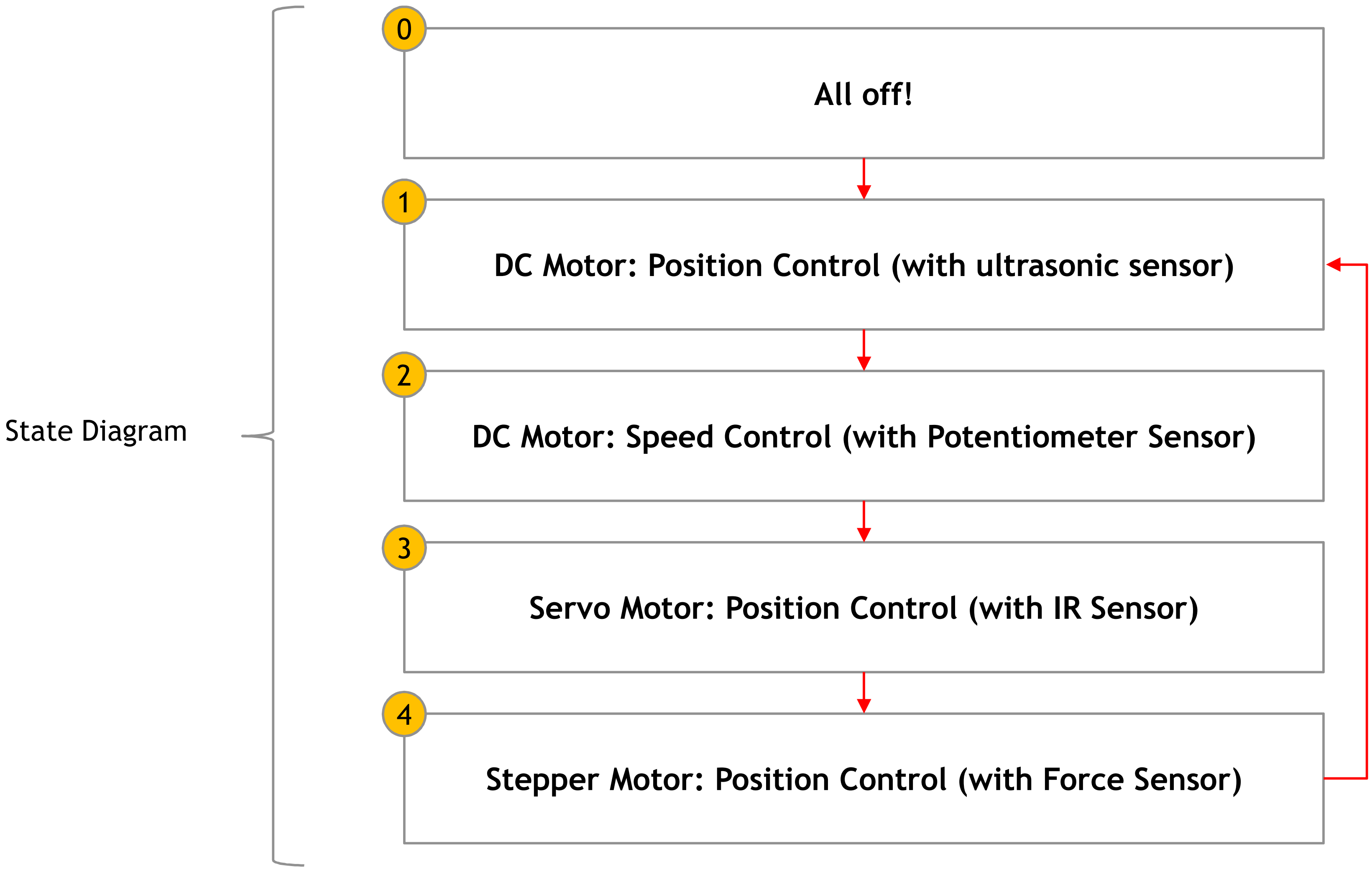
Task 7: Sensors and Motor Control Lab

12th October 2017

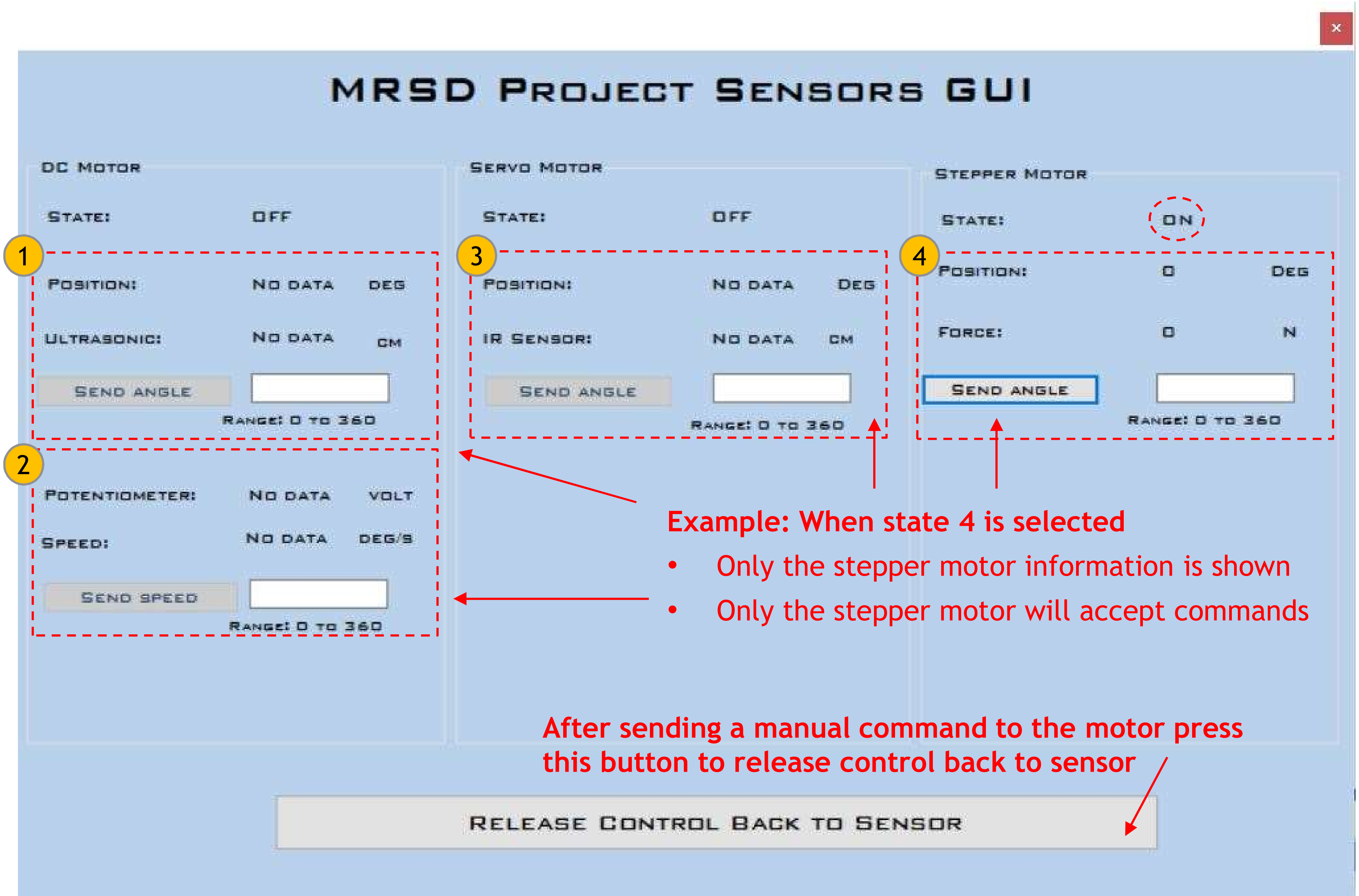
# We have opted for a simple design with extra garbage handling features for enhanced robustness



The touch-button in the breadboard will cycle through the different state diagram of the prototype



The Graphical user interface (GUI) intuitively guides the user on what controls are available for the current state you have selected

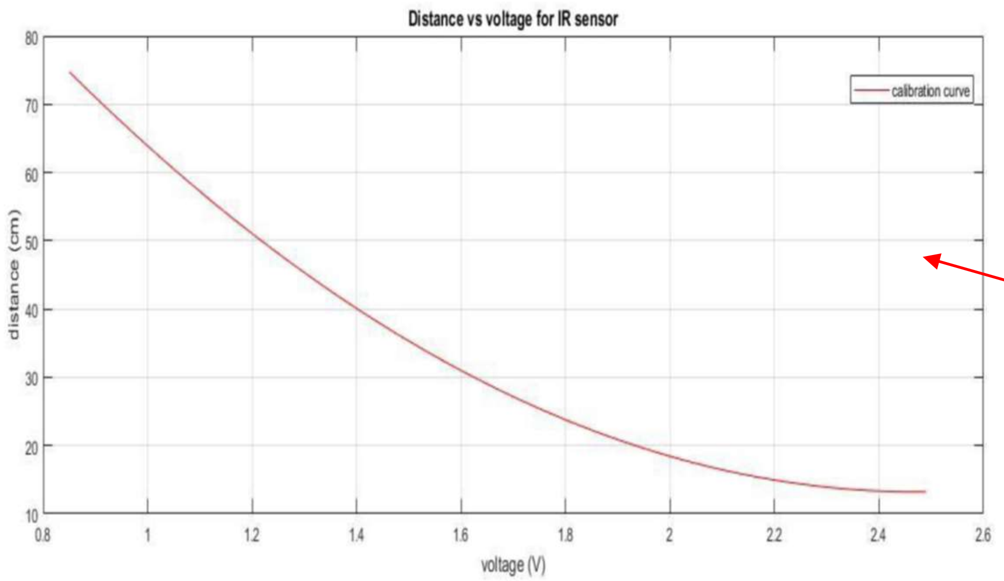


*Example*

# The sensors were calibrated in order to be able to control the motors

**Sharp IR proximity sensor ( GP2Y0A21YK, 10-80 cm)**

The transfer function plot for the sharp IR proximity sensor is shown below




For this working range of the sensor, the relation between voltage and the distance in cm is given by the following equation,

$$IRdistance = 23.4\,(IRvolt)^2 - 115.7\,(IRvolt) + 156.2$$

A different calibration curve was made for each of the motors/sensors

- **Press the pushdown button to cycle through the state diagram**
- **For each state (other than 0)**
  - Confirm that control of a motor will be handed by default to a sensor
  - Try sending a command to a motor for disabling the sensor control and perform manual control
  - Give the command back to the sensor by cycling through the different states or pushing “Release command to sensor”
  - Try sending several garbage commands to test the resilience of the interface
- **Test for the DC Motor (state 1)**
  - Interact with the ultrasonic sensor to transform distances measured in centimetres into a DC motor position in degrees
  - Send target position command to the DC motor (0 to 360 degrees in integers)
  - Monitor the evolution of the angle until the DC motor reaches the target position
- **Test for the DC motor (state 2)**
  - Interact with the potentiometer to transform applied voltage to DC motor speed
  - Send target speed command to the DC motor (0 to 360 degrees per second in integers)
  - Monitor the evolution of the angular speed until the DC motor reaches the target speed
- **Test for the Servo motor (state 3)**
  - Interact with the IR sensor to transform volts into a Servo Motor position in degrees
  - Send target position command to the Servo motor (0 to 360 degrees in integers)
  - Monitor the evolution of the angle until the Servo Motor reaches the target position
- **Test for the Stepper motor (state 3)**
  - Interact with the Force Sensor to transform pressure into a Stepper Motor position in degrees
  - Send target position command to the Stepper motor (0 to 360 degrees in integers)
  - Monitor the evolution of the angle until the Stepper motor reaches the target position

- **All in one interface to access all sensors, motors and push-down buttons**
  - Micro-controller based architecture
  - Push-button with de-bouncing to cycle through states
  - RC servomotor, DC motor with encoder, and stepper motor
  - Ultra-sonic sensor, potentiometer, IR sensor and Force sensor
- **Single wood board with all hardware**
  - All components mounted in a single wooden board
  - All components tightly connected to the wooden board
  - Easy to track movement of the motors
- **Integrated and intuitive interface (GUI)**
  - Written in C# for fast processing and quick development
  - Intuitive with relevant information clustered together
  - Simple with all non-relevant information switched off
  - Protects you from sending the wrong command disabling non-relevant buttons
- **Robust (does not breakdown)**
  - No garbage controls sent to the hardware
  - Garbage incoming from the hardware filtered out

# BACKUP: FLYSENSE CORE PROJECT FLASH REPORT

| Tracks | Milestones for PR 1 - 20th Oct 2017 | Internal tasks to deliver on PR 1 | Achievements since CoDR was written | Risks/Next steps needed to deliver on PR1 |
|---|---|---|---|---|
| **AR** | • First visualization of sensor suite | • First visualization of Heads-up display with fake sensor data | • Air Lab Hololens borrowed being for testing (zoom-in) | • Test other AR headsets Finalize mock-up of Heads-up Display |
| **UI/UX** | • First draft | • First draft of UI/UX. | • Analyzed the performance of Hololens in hand tracking, voice tracking and voice recognition identified gaps with against the required "ideal" performance | • Meet with CMU experts<br>– Jean Oh tomorrow at 15:00<br>– Aaron Steinfeld next week<br>– Richard Stern pending confirmation |
| **Sensing** | • Point cloud of<br>• Lidar Data set | • Use data from Lidar data set to create and visualize point cloud. | • Lab LIDAR borrowed for preliminary testing (mapping functionality is needed - zoom in) | • NEA cloud points to be retrieved tomorrow at 15:00 |
| **System Integration & Testing** | | | • Jetson TX1 IMU available for testing but now using laptops | • Integrate components as testing is finalized (not in critical path) |
| **Project Management** | • Procurement | • Procure IMU, GPS, LIDAR and Onboard computer | • Double-sided NDA swith NEA<br>• Sponsorship about to be signed with NDA | • Secure access to Velodyne |

Source: CoDR, team update

| Hololens testing: Pros | Hololens testing: Cons |
| --- | --- |
| • Easy to build applications<br>• Excellent head tracking (gaze)<br>• Resilience to lighting conditions (overlaid reality)<br>• Excellent 3D Audio (with no background noise)<br>• Excellent Voice Commands (with no background noise)<br>• Works both in wireless and wired mode (voice commands only work online) | • Feels heavy after using it by 30/40 minutes<br>• The headset can move in the head and the hologram loses the correct alignment<br>• Hand gestures tracking does not work well enough to interact with tool<br>• Voice commands do not work with helicopter background noise – tested in Robolounge)<br>• Voice commands only work with online connectivity<br>• Limited field of view (-30 to 30 degrees) |