

# Romotive Romo Control

Daniel Volz

# Proposal

For my final project, I will develop an iPhone-only app that combines the iPhone's sensors and Romotive's SDK to control the Romotive Romo. For speed control, the iPhone's tilt can be measured using the accelerometer to determine the velocity of the Romo. Using the iPhone's GPS the Romo will travel towards a user-specified destination and stop once it has been reached. Once a destination has been reached, the iPhone will take a picture to verify that it is in the specified location. Additionally, the Romo's facial emotion will change to visually display its success. Rather than a location, the Romo can be told to travel in a particular direction. The magnetometer will be used to tell the Romo to head in a user-specified direction. Finally, the Romo will have an alarm feature that will flash the Romo's LED and make sound at a specified time.



# Parts of iPhone Used

- \* Accelerometer
- \* Magnetometer
- \* GPS
- \* Camera
- \* Speaker

# Parts of Romo Used

- \* Motors
- \* LED
- \* Facial expressions

# Parts of iOS Used

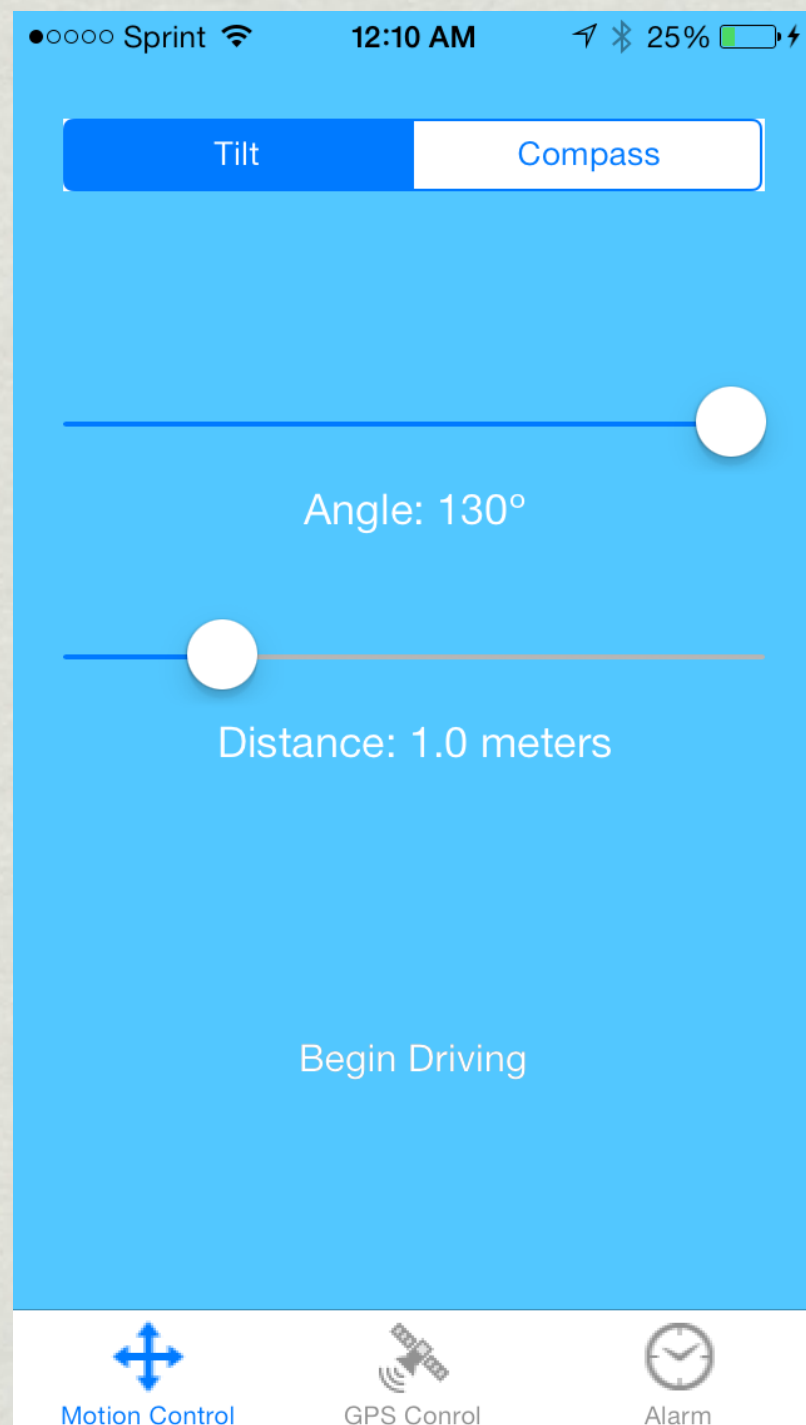
- \* UITabBarController for choosing between Romo modes (not shown in demonstration screens)
- \* UISlider for controlling speed of Romo
- \* UISegmented Control for choosing movement mode
- \* MKMapView for GPS control
- \* UIImagePickerController for camera control
- \* CLLocationManager for compass control
- \* AVAudioPlayer and NSTimer for alarm



# Parts of RomotiveSDK Used

- ✱ RMCharacter for changing the Romo's facial expression
- ✱ RMCoreLEDs for controlling robot's LEDs
- ✱ RMCoreMotor for controlling Romo movement
- ✱ RMCore for general robot control and initialization

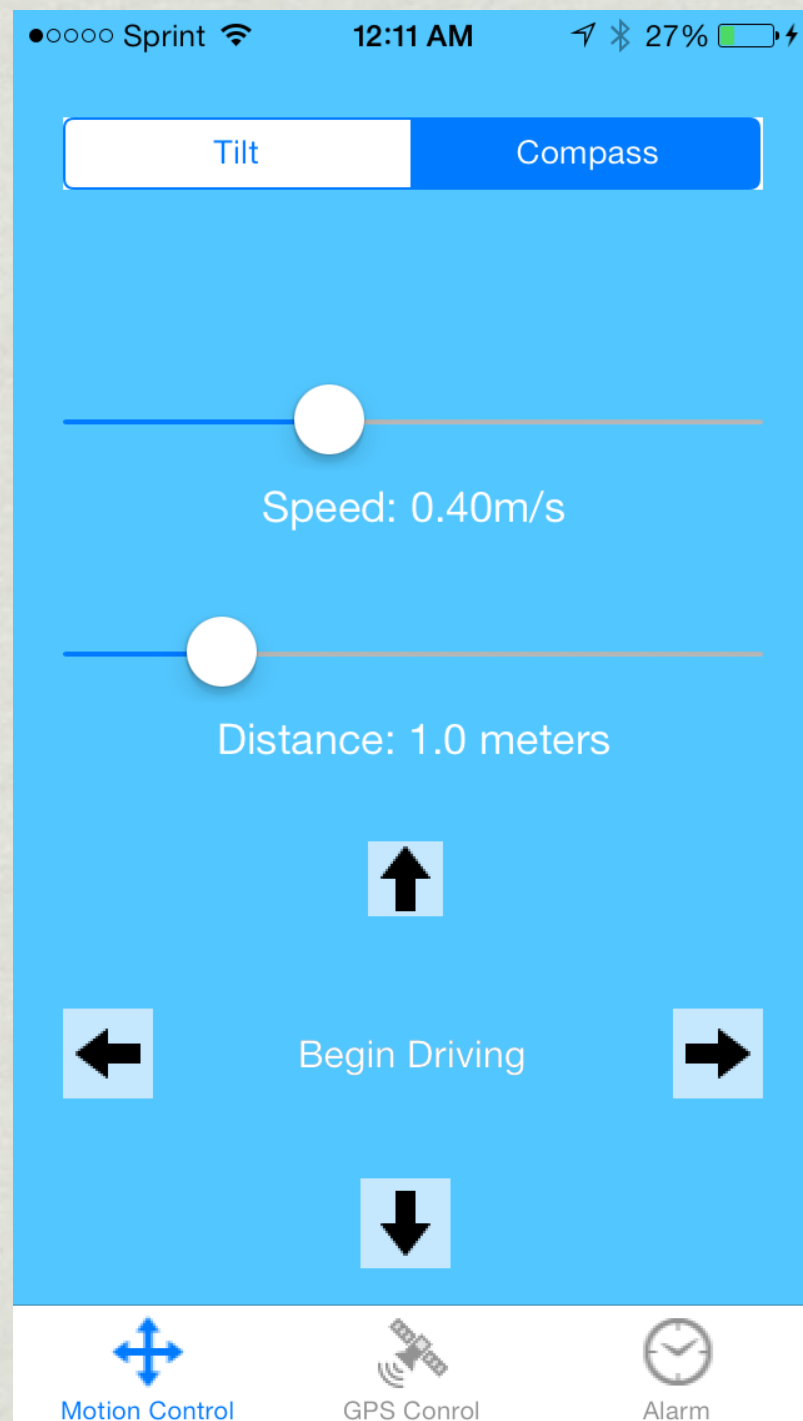
# Distance and Direction Control



- \* Angle control slider determines speed and angle
- \* Accelerometer is measured to determine angle
- \* Distance control slider

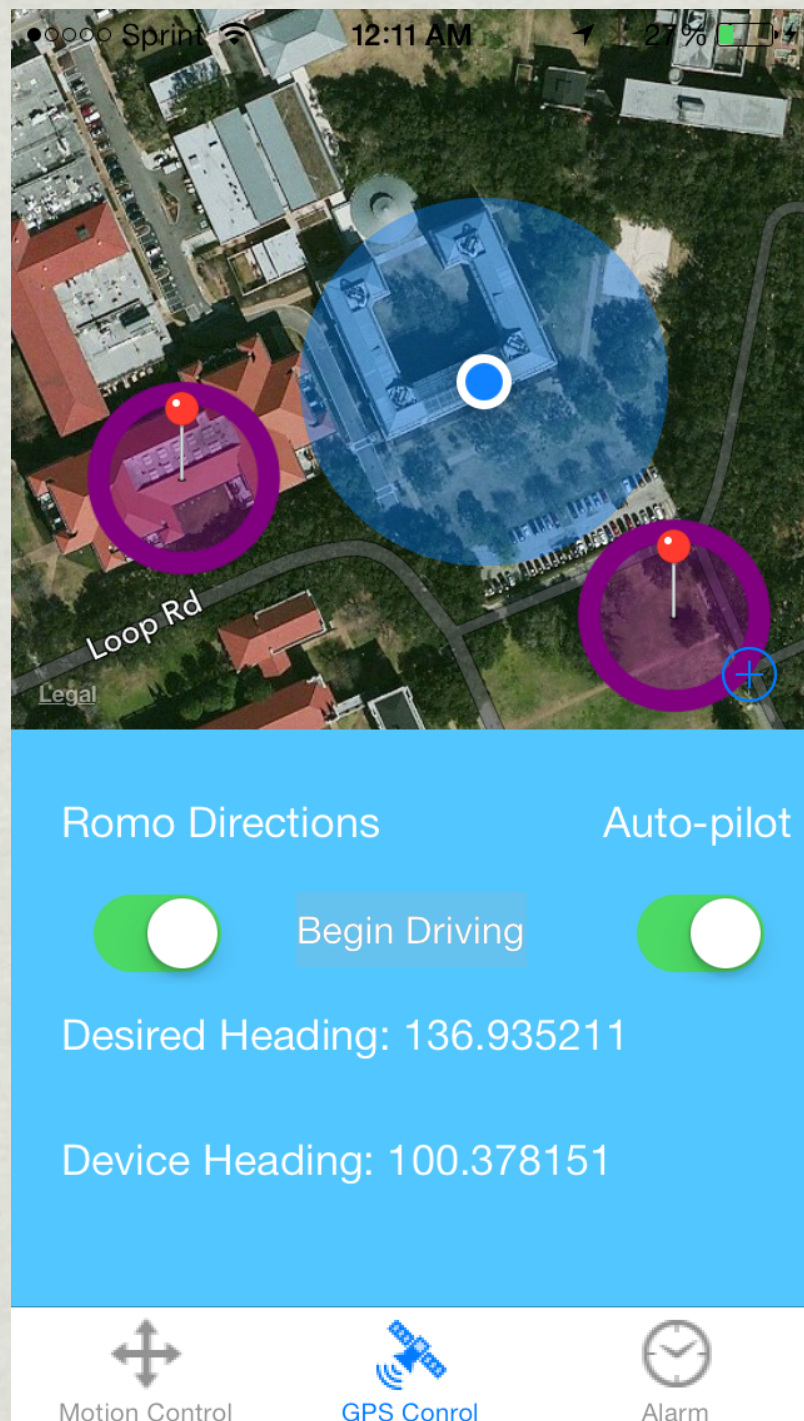


# Distance and Direction Control



- \* Speed control slider
- \* Distance control slider
- \* Direction control using compass

# GPS Control



- \* Select a point
- \* Romo will head towards most recently dropped waypoint
- \* Automatically remove waypoint once destination reached
- \* Continue to next waypoint
- \* Auto-pilot allows Romo to correct heading if it detects it's going the wrong direction
- \* Romo Directions changes between Romo API heading calculation and manual calculation

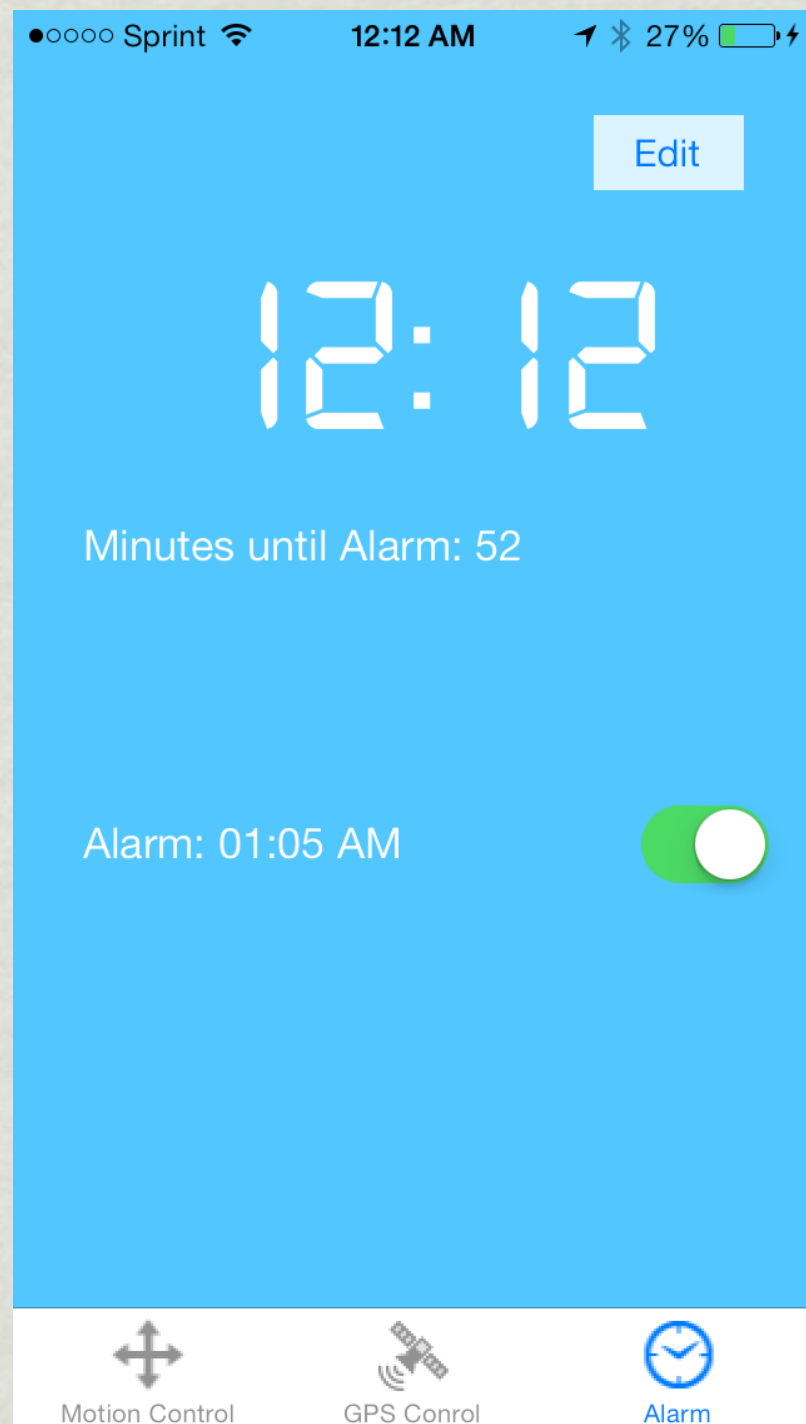


# Destination Reached



- \* Take a picture at destination
- \* Upload photo to iCloud
- \* Either GPS location reached or certain distance travelled
- \* Change Romo facial expression
- \* Swipe up to dismiss

# Alarm Clock



- \* Sound alarm at specified time
- \* Tell user how much time until alarm goes off
- \* Flash Romo LED at user-specified time
- \* Romo spins in circle (maybe dangerous if on nightstand)