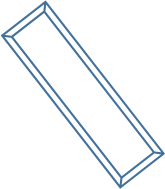
AE-35 Antenna AZ/EL Rotor System



l

*I've just picked up a fault in the AE35 unit. It's going to go 100% failure in 72 hours. – HAL 9000*

Table of Contents

[Introduction 3](#_Toc471043481)

[Basic Operation (standalone) 4](#_Toc471043482)

[Software 5](#_Toc471043483)

[Overview 5](#_Toc471043484)

[State Machine 6](#_Toc471043485)

[Cyclic Executive 7](#_Toc471043486)

[Gimbal Object 8](#_Toc471043487)

[Orbital Tracking 8](#_Toc471043488)

[Hardware 8](#_Toc471043489)

[CPU 8](#_Toc471043490)

[Gimbal Drive 12](#_Toc471043491)

[Software 13](#_Toc471043492)

# Introduction

This is a prototype elevation over azimuth antenna pointing system for tracking low earth orbiting satellites. It is built around the *qrpTracker* library (Bruce Robertson VE9QRP) which is in turn derived from *PLAN13* (James Miller G3RUH).

In standalone mode; the system is capable of tracking a handful of orbiting objects compiled into its Arduino sketch. As a consequence of single precision arithmetic, its simple perturbation model ages rapidly. The two line element orbital descriptions should be updated and the sketch recompiled and downloaded before each use.

The system is also designed to obey serial commands using the EASYCOMM I protocol. The serial command functionality has not been tested.

The central processing unit is built around an Arduino 2650 Mega R3.

The gimbal system was cannibalized from an automatic satellite TV antenna and provides stepper motor drives for a lightweight elevation over azimuth pointing system with a +/-180o azimuth range and a 50o maximum elevation.

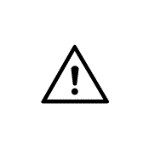
The primary antenna is an Alaskan Arrow 145Mhz/440Mhz crossed Yagi-Uda with four 2m elements and thirteen 70cm elements.

The secondary antenna is an offset feed Ku band parabolic reflector. No feed horn is installed at this time.

There are two four-foot outrigger truss assemblies with 1” cylindrical and 1” square tubing for mounting antenna elements.

# Basic Operation (standalone)

There are two SPST toggle switches with safety covers which control power to the central processing unit and the drive servos. Lift the safety covers and toggle both to UP=ON.

Never operate the unit with only CPU power to prevent the motor drive from overloading the Arduino power regulator.

The lighted LCD module displays the current state and provides a six button input panel to set initial parameters, advance thru the operating modes and manually drive the antenna gimbal.

The first step is to update the Arduino sketch keps[] string array with current two line elements for the orbiting objects to be tracked and verify the hard coded observer latitude and longitude.

The user will ❷ input the correct GMT time and date and ❹ steer the gimbal system until the antenna is pointing to true north. The gimbal position is displayed raw motor counts until the initial orientation is determined.

The system will ❺ determine the home position in elevation and then ❻ tracking mode can be selected. The automatic track mode can be overridden at any point using the left/right/up/down inputs to ❹ drive the antenna.



SET TIME (GMT)

23:25:03

❷CONFIG

aE-35 Antenna

DEC 28 2016

❶INIT

DRIV 23:12:12.1

A00000E00000

❹DRIVE

HOME 23:12:16.6

A00000E02100

❺HOME

TRAC 23:13:40.6

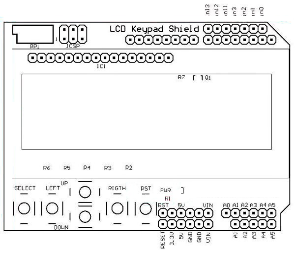
SO-50AZ24.2 E:5.0

❻TRACK

SET DATE 12-23-2016

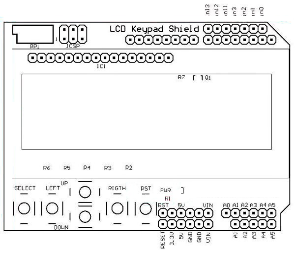
SET SATELLITE

SO-50



STATE HH:MM:SS.M

SAT AZ:000 EL:000



**DECREMENT**

**CURSOR LEFT**



**INCREMENT**

**CURSOR RIGHT**

**SELECT**

**RESET**



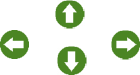
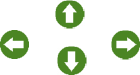
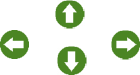
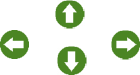
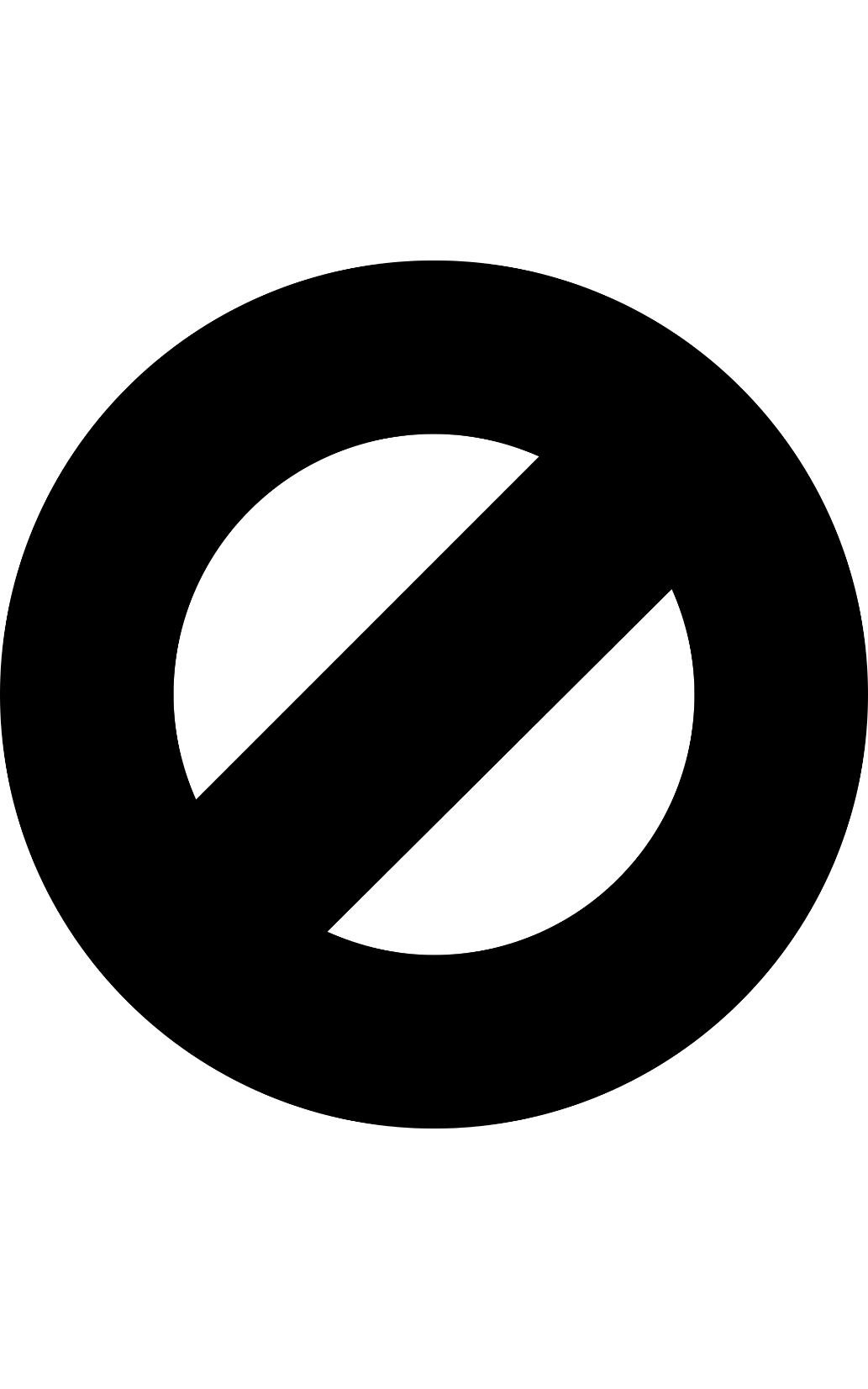
DRIV 23:15:SS.5

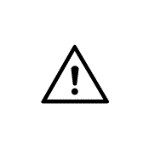
AZ:24.2 E:5.0

<ORBITAL PARAMS>

❸GPS LOCK

MAG HEADING:31.8 ROLL+2PITCH-12



The motors are idled if the commanded position has not changed for 1.5 seconds. It is recommended to manually park antenna off-axis if the LEO is below the horizon for an extended period of time to prevent overheating L293D drive circuits.

# Software

## Overview

Loop()

Cyclic Executive

\_state

Real Time Functions

State Machine

\_state

buttons & switches

serial input

\_state

s

Antenna Gimbal Object

target Az/El

Motor Position(s)

Plan13 library

time and date

orbital elements

target Az/El

Accelstepper library

Motor Position(s)

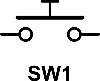
## State Machine

The software implements a basic state machine with a table of entry point functions, executed once when the state transition is invoked, and an array defining state transition conditions.

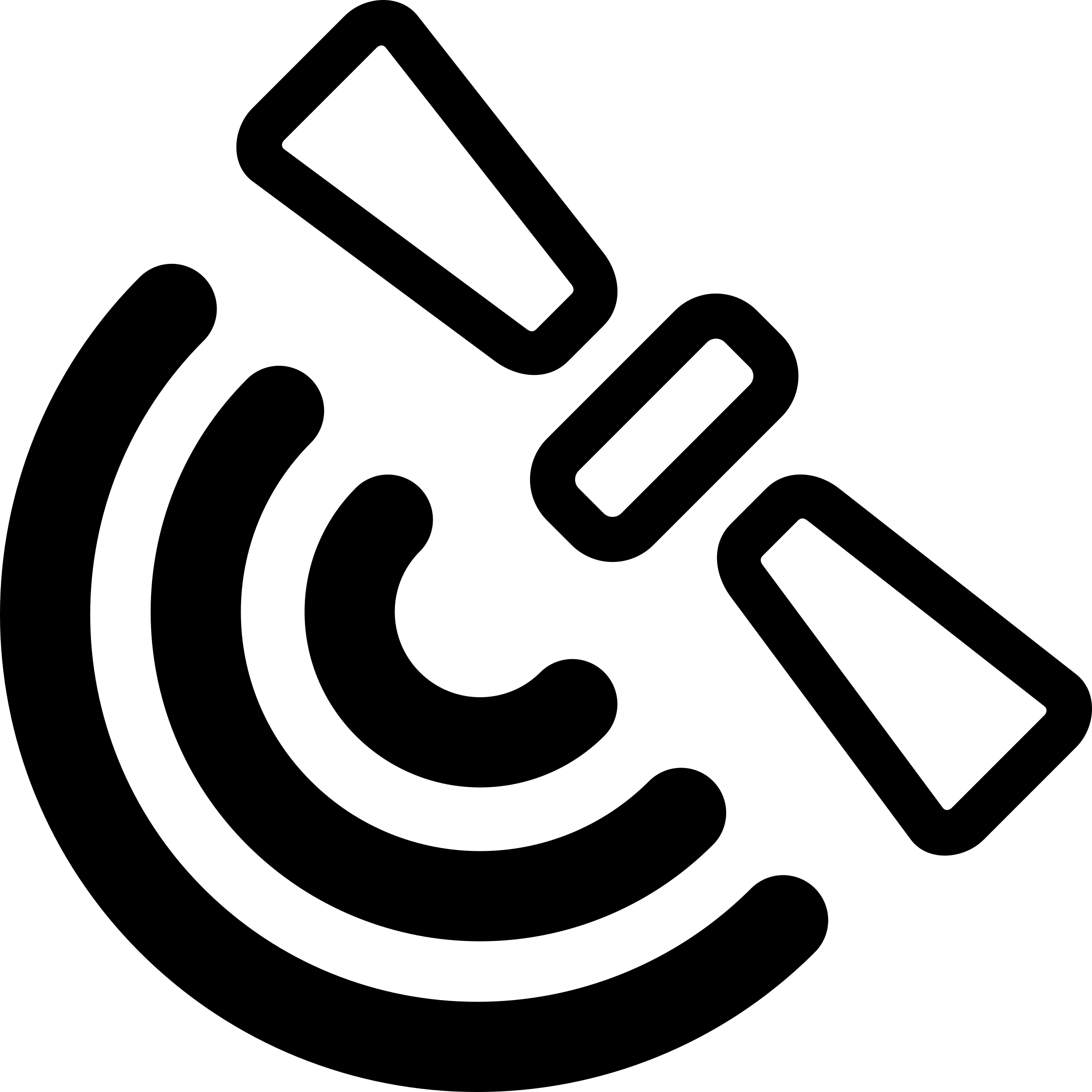
State transitions are invoked from user SELECT button input, LIMIT switches, SERIAL input, or GPS LOCK.

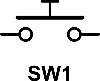
GPS

LOCK

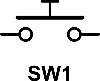


SEL





SEL



SEL

HOM

Zero gimbal position against limit stops

INIT

1. Initialize
2. Input date
3. Input time

TRAC

Automatic antenna drive

DRIVE

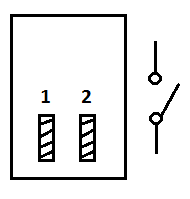
Manual antenna drive

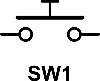
GPSL

1. Read compass
2. Read accel.
3. Read GPS

CNFG

1. Input LEO
2. Parse TLE
3. Init PLAN13





UP | DOWN |

RIGHT | LEFT

LIMIT

## Cyclic Executive

The sketch loop() entry point implements a basic cyclic executive which manages the state machine and calls four sets of functions at regular intervals based on the current state

Each state defines an entry point and four sets of executive functions to be called at various intervals.

StateFunctionsType

ExecutiveFunctionsArray

A set of functions to be executed ALWAYS (as frequently as possible)

A set of functions to be executed FAST (50 times a second)

A set of functions to be executed MEDIUM (20 times a second)

A set of functions to be executed SLOW (10 times a second)

Entry point

l

transition condition?

Execute StateFunctionsType entry point function (once)

Execute all ALWAYS ExecutiveFunctionsArray for current \_state

Execute all FAST ExecutiveFunctionsArray for current \_state

20ms elapsed?

Execute all FAST ExecutiveFunctionsArray for current \_state

50ms elapsed?

Execute all FAST ExecutiveFunctionsArray for current \_state

100ms elapsed?

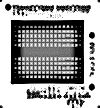
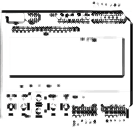
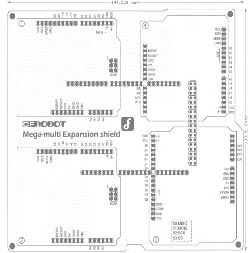
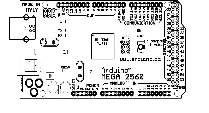
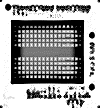
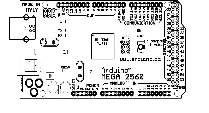
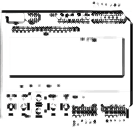
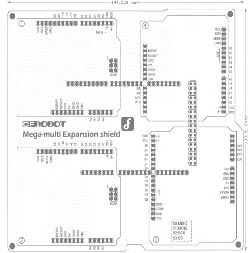
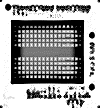
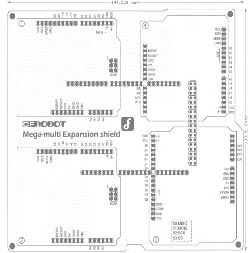
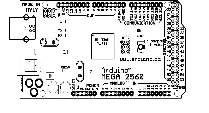
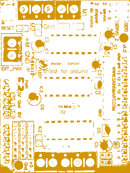
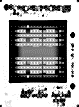
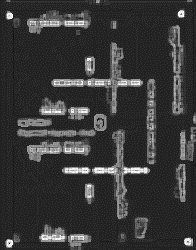
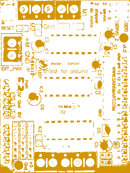
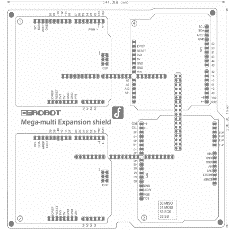
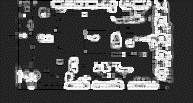
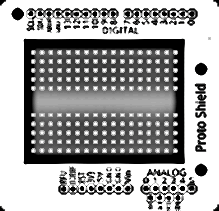
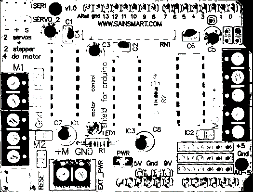
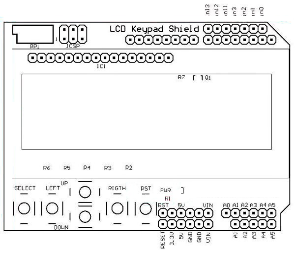
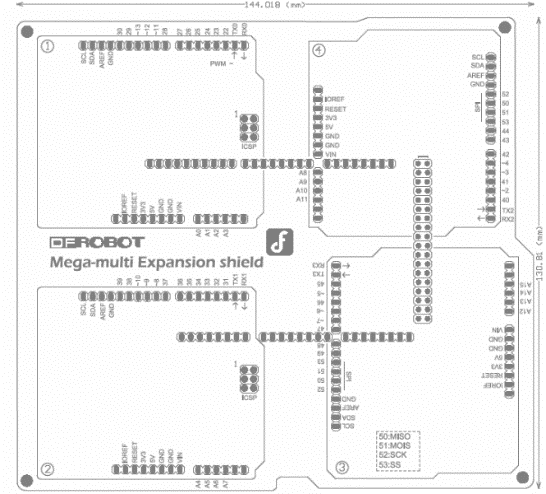
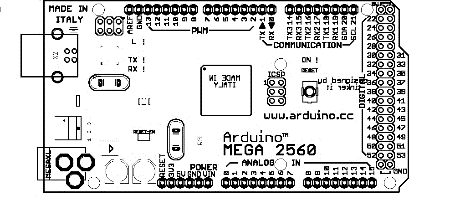
## Gimbal Object

The heart of the system is an object class which defines the antenna gimbal mechanism. This object encand contains all the physical limits, coordinate transformations from gimbal head to plumb-north, and relationship between motor steps and angular movement.

## Orbital Tracking

# Hardware

## CPU

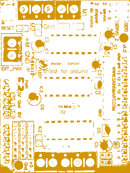


L293D Stepper Motor Shield

‘

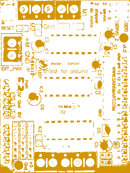
Arduino Mega 2560

Multi Arduino Mega Expansion Shield

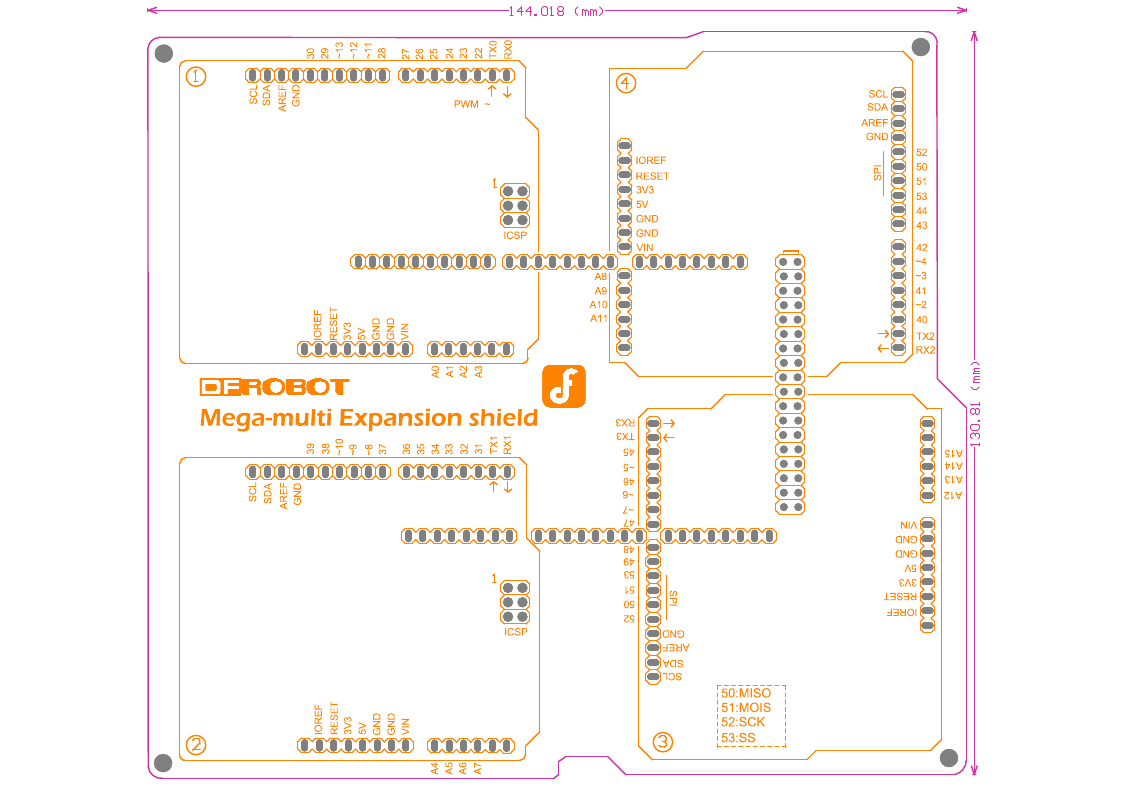


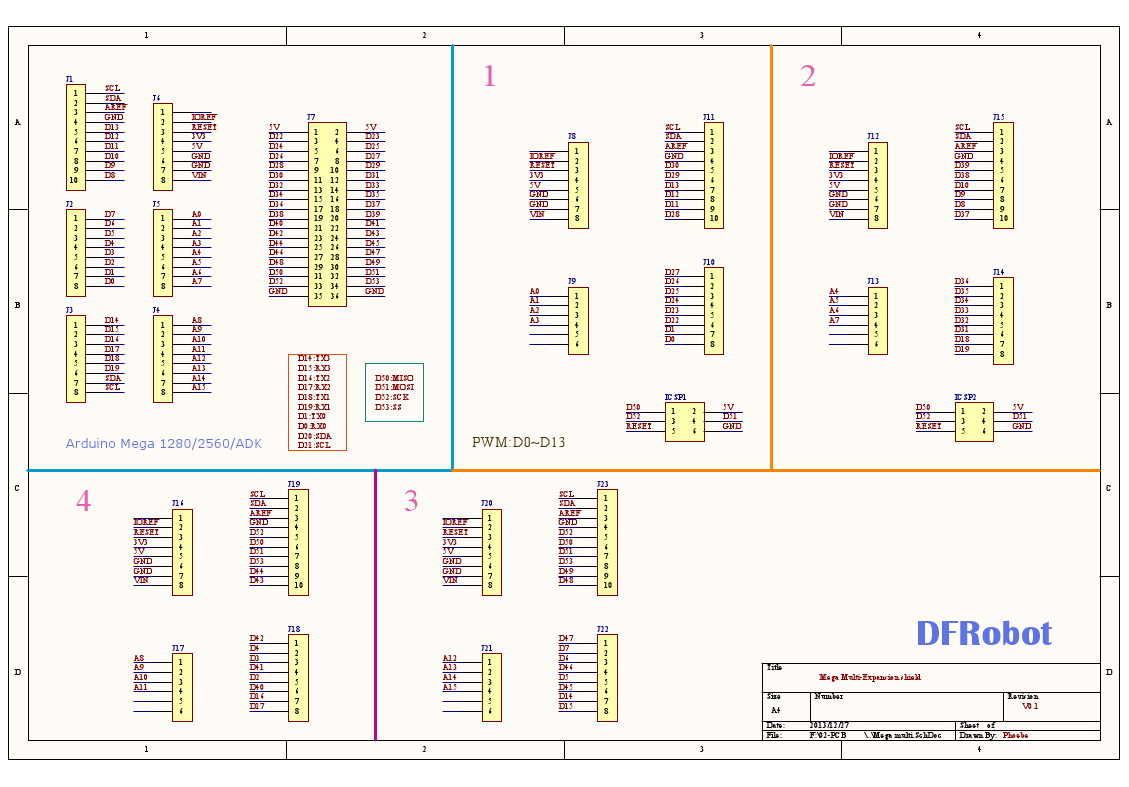
1602 LCD Keypad shield

|  |  |  |
| --- | --- | --- |
| Arduino 2560 Mega R3 | | |
| DFRobot Multi Mega Expansion Board | | |
|  | 1602 LCD Keypad Shield | |
|  | L293D Stepper Motor Shield | |
|  | Prototyping Shield + mini breadboard | |
|  |  | MPU 9150 Accelerometer/Magnetometer |



Prototyping shield + mini breadboard





## Gimbal Drive

# Software



SET TIME (GMT)

23:25:03

❷CONFIG

aE-35 Antenna

DEC 28 2016

❶INIT

DRIV 23:12:12.1

A00000E00000

❹DRIVE

HOME 23:12:16.6

A00000E02100

❺HOME

TRAC 23:13:40.6

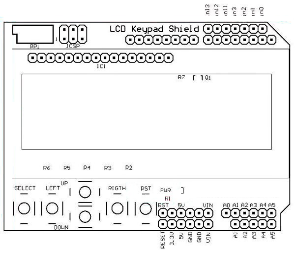
SO-50AZ24.2 E:5.0

❻TRACK

SET DATE 12-23-2016

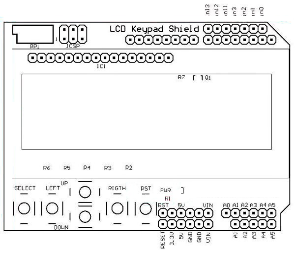
SET SATELLITE

SO-50



STATE HH:MM:SS.M

SAT AZ:000 EL:000



**DECREMENT**

**CURSOR LEFT**



**INCREMENT**

**CURSOR RIGHT**

**SELECT**

**RESET**



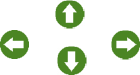
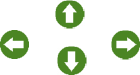
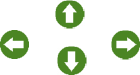
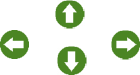
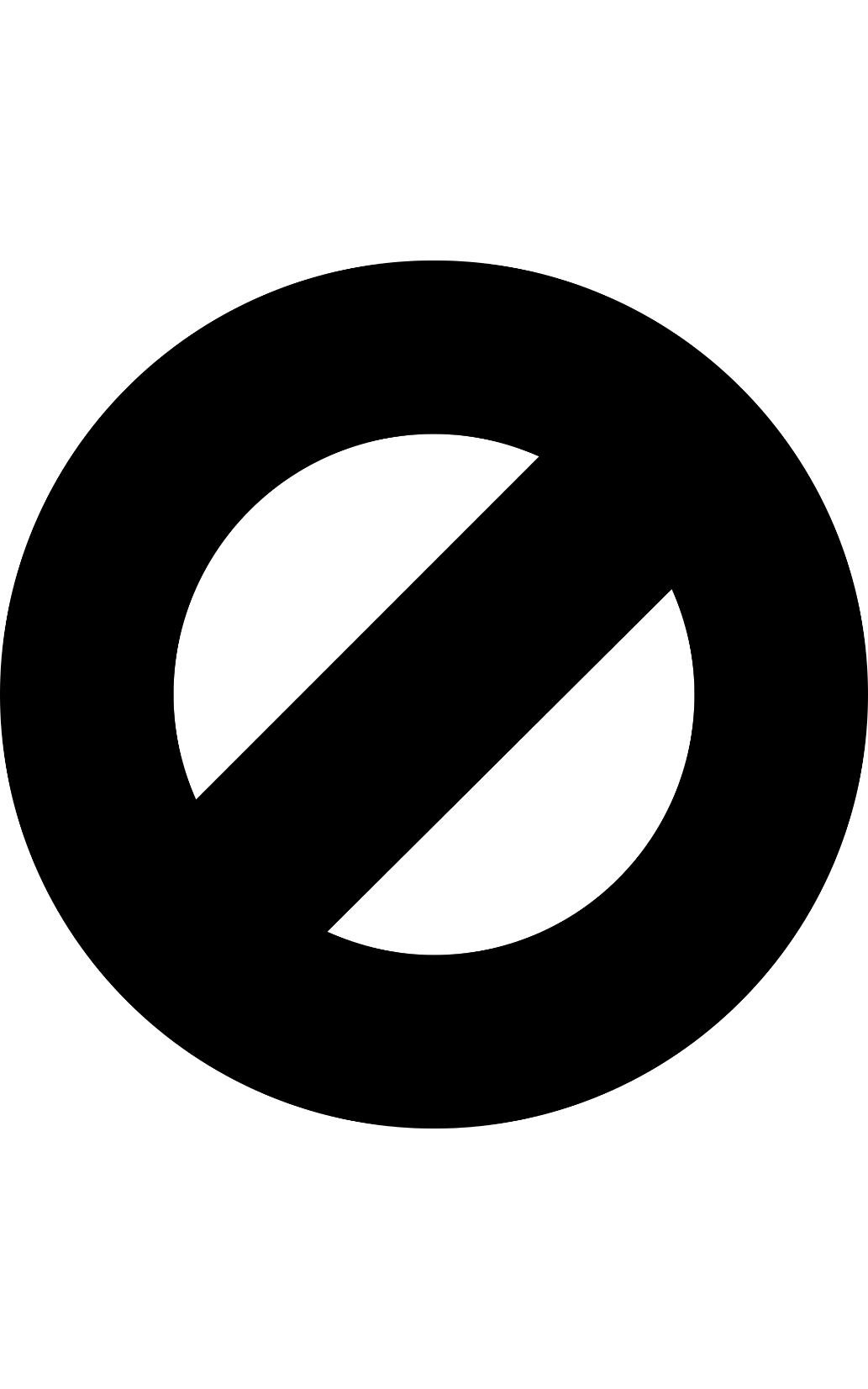
DRIV 23:15:SS.5

AZ:24.2 E:5.0

<ORBITAL PARAMS>

❸GPS LOCK

MAG HEADING:31.8 ROLL+2PITCH-12



,

