SubArc V3 Specifications

Customization: SubArc can be easily customized to fit needs of higher/lower resolution and high/lower accuracy all by changing the magnet count of the ring and number of hall sensors in the array on the readhead. The following equation determines needed custom specifications.

minimum slope value of d/dθ (periodic function) × 2π (period) × |maximum field strength| × MFC factor × # of magnet poles ÷ repeatability range of hall sensor = CPR or resolution

Customization Parameters

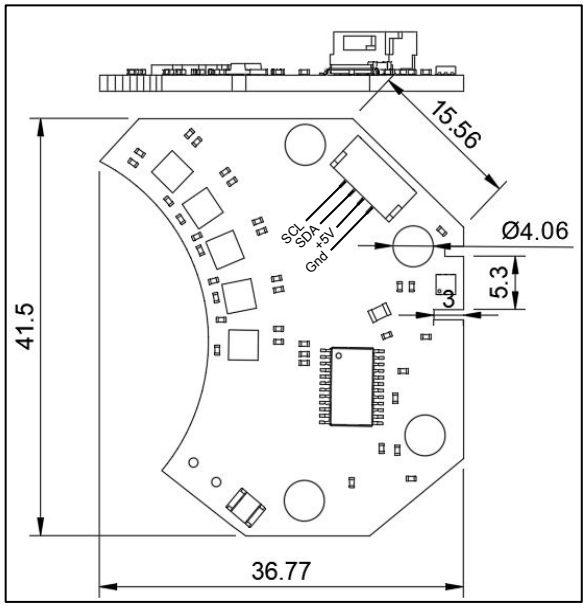
- **M** = Magnet count
- **A** = # of hall sensors in the array
- **B** = bore hole diameter in mm
- **Sensor name:** SubArc-[M]-[A]-[B]

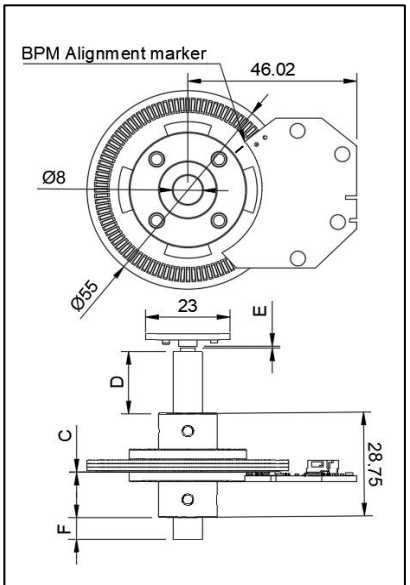
Mounting Parameters

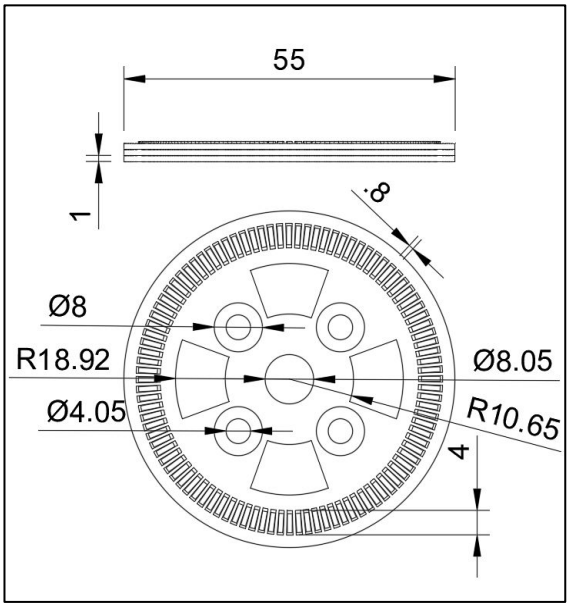
- **F & D** = any MPM ring mounting distances possible within reason
- **E** = AS5600 BPM subdivider ride height ~0.5mm
- **C** = Main readhead ride height 0.05mm-0.2mm

Standard SubArc-100-5-8 Specifications:

- **Resolution:** Maximum calibration = 20-bits
- **Average Accuracy:** 0.0384° (139 arcseconds) (can be better or worse depending upon mounting eccentricity)
- **Readhead height:** 0.05mm - 0.2mm
- **Communication:** I2C, 64 possible encoders/microcontroller
- **Disk space needed:** 250MB/encoder
- **Frequency:** Max safe clock speed = 400 kHz, Max reliable data speed (20-bit) = 110 Hz
- **Mass:** 48g
- **Tolerance:** H7, 8mm bore
- **Temperature range:** Maximum calibration = -20℃ to 50℃ (-4℉ to 122℉)
- **Ring diameter:** 55mm
- **Operative voltage:** 3.3V





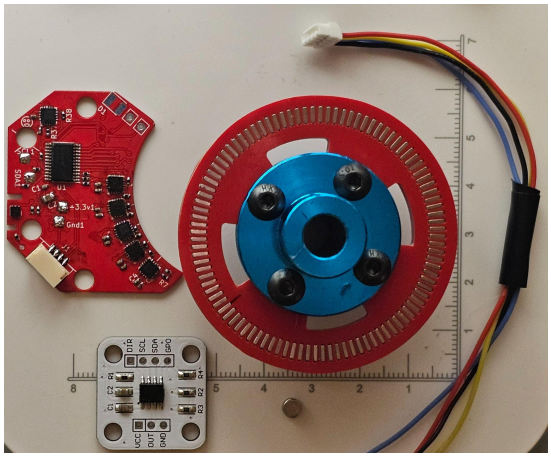


SubArc V3 Installation Guide

Installation: All referenced files can be found on the GitHub. To implement multiple encoders into a larger robotics framework of choice, the code can easily be modified to fit in Python.

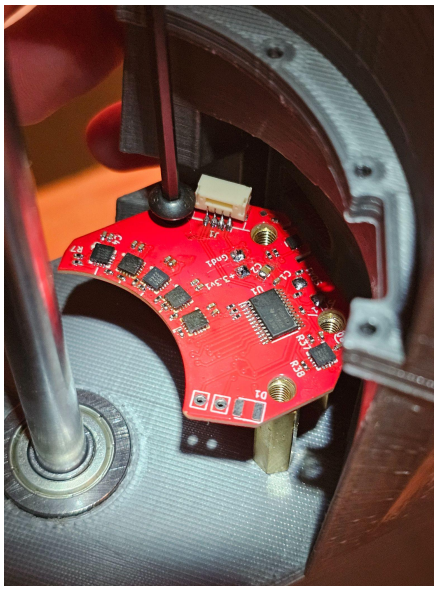
Step 1:

- Ensure parts came including:
- 1x AS5600 subdivider
- 1x BPM
- 1x SubArc readhead
- 1x SubArc magnet ring
- 1x Microcontroller
- 2 x I2C wiring cable (one with JST connector adapter)
- 1x Computer (Jetson nano, raspberry pi, laptop, PC, etc.)
- 4x M4 8mm screws



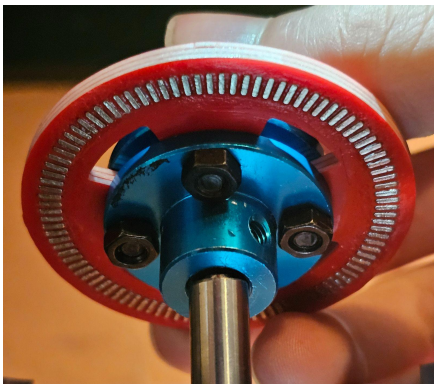
Step 2:

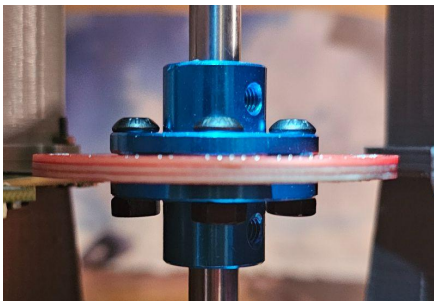
- Use 4x M4 8mm screws to Mount SubArc main readhead
- Ensure readhead can't move and will not vibrate
- Plug in JST power & communication cable



Step 3:

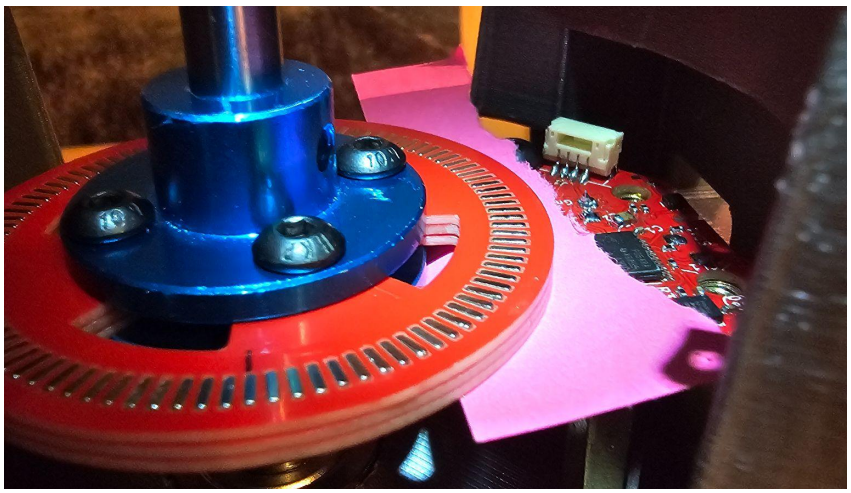
- Place SubArc ring onto shaft
- Adjust flange couplers if necessary, tolerance should be > h7
- Fit should be friction tight





Step 4:

- Tune readhead height with known width slim object with < 0.2mm thicknesses
- Sticky Notes or average printer paper is a good option
- Ensure height is exact thickness by friction fit with leveling object

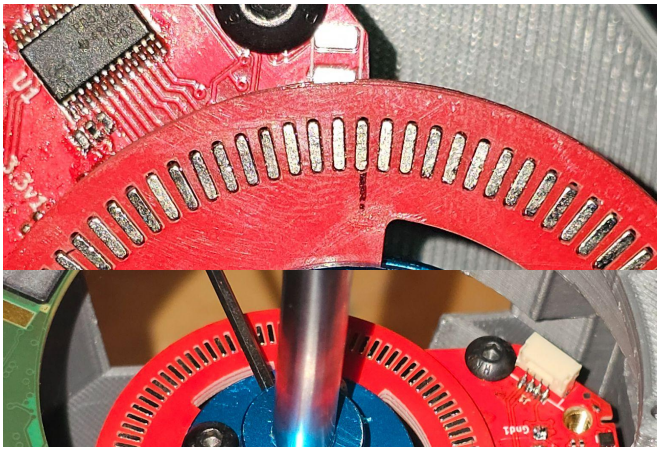


SubArc V3 Installation Guide

Note: The encoder will not operate properly or at all if installation tolerances are not met. Additionally, a soldering iron is needed to solder on pins of AS5600 subdivider.

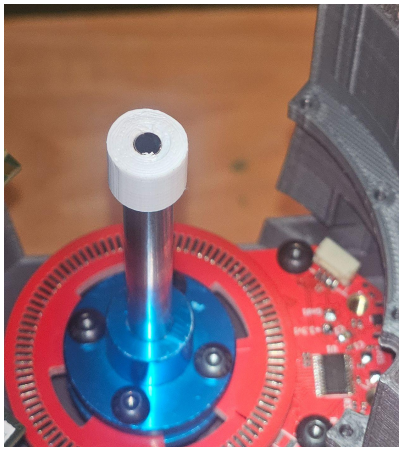
Step 5:

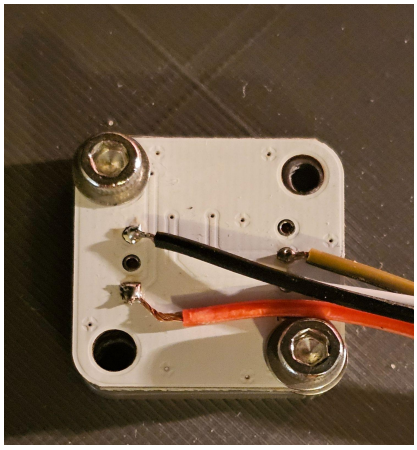
- Align zeroing marker on SubArc ring with edge near light indicator
- Should be less ½ magnet thickness of accuracy (~ +-1mm)
- Once zeroed, tighten set screws while making sure readheight stays constant and the marker stays aligned with less than ½ magnets width of accuracy



Step 6:

- Set BPM on shaft using aligning tool included, mount AS5600 directly on top with space to tune BPM rotation
- Connect AS5600 I2C and power to microcontroller and read absolute position using included script
- Tune magnet rotation till 0 +- 0.5° is achieved
- Glue in magnet on shaft





Step 7:

- Connect SubArc to 2nd I2C port or wire in parallel with AS5600 I2C
- Load "SubArcV3 Microcontroller" and upload and test code to ensure an AS5600 value < 0 +-0.5°, and that the SubArc readhead producing temperature and all magnetic field values
- Ensure Serial command communicates properly with microcontroller

Step 8:

- Download specialized calibration data files for your encoder, and import them into the main python file folder
- Run the main "SubArcV3LiveAnglePosition.py" to return live angular position
- The main python file can be modified to read multiple encoders or operate on a larger script