Design

The design distinguishes two ‘sides’: user and system. The user is the side where the api is defined that provides access to the rover functions. It contains operations like:

* getPosition()
* getLineSensor()
* getCollisionSensor()
* setTrackSpeed()
* setTrackDirection

From user to system:

Sys\_U2S()

{

While(true) {

Sem\_take(u2s\_req);

// Copy pwm from memory to IO

Sem\_give(u2s\_ack);

}

}

Usr\_U2S\_setPwm(in dutycycle)

{

// Copy duty cycle to memory

Sem\_give(u2s\_req);

Sem\_take(u2s\_ack);

}

From system to user:

Sys\_S2U()

{

While (true) {

Sem\_take(s2u\_req);

// Copy sensors to memory

Sem\_give(s2u\_ack);

}

}

Usr\_U2S\_getLine(out line)

{

Sem\_give(s2u\_req);

Sem\_take(s2u\_ack);

// Copy line sensor from memory

}

System side:

On the system side, the Sys\_ functions run that handle interaction with user side. Furthermore, ISR’s are installed on state changes on the encoders.

Quadrature decoding

AUp()

{

if B=0 then pos++

if B=1 then pos--

}

ADown()

{

If B=0 then pos--

If B=1 then pos++

}

BUp()

{

if A=0 then pos--

if A=1 then pos++

}

BDown()

{

If A=0 then pos++

If A=1 then pos--

}