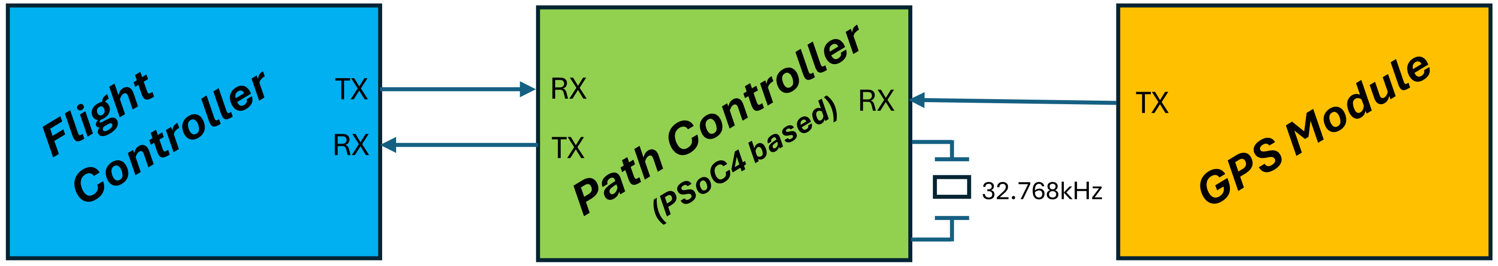
Interface to Drone Path Controller:

The physical interface will consist of a Cypress PSoC4 uController utilizing two UART peripherals. One full-duplex UART will be used to communicate commands/status between the Flight Controller and the Path Controller.   A second half-duplex UART will be used to receive packets from the GPS module. Both UARTs run at 115,200 baud using 8N1.



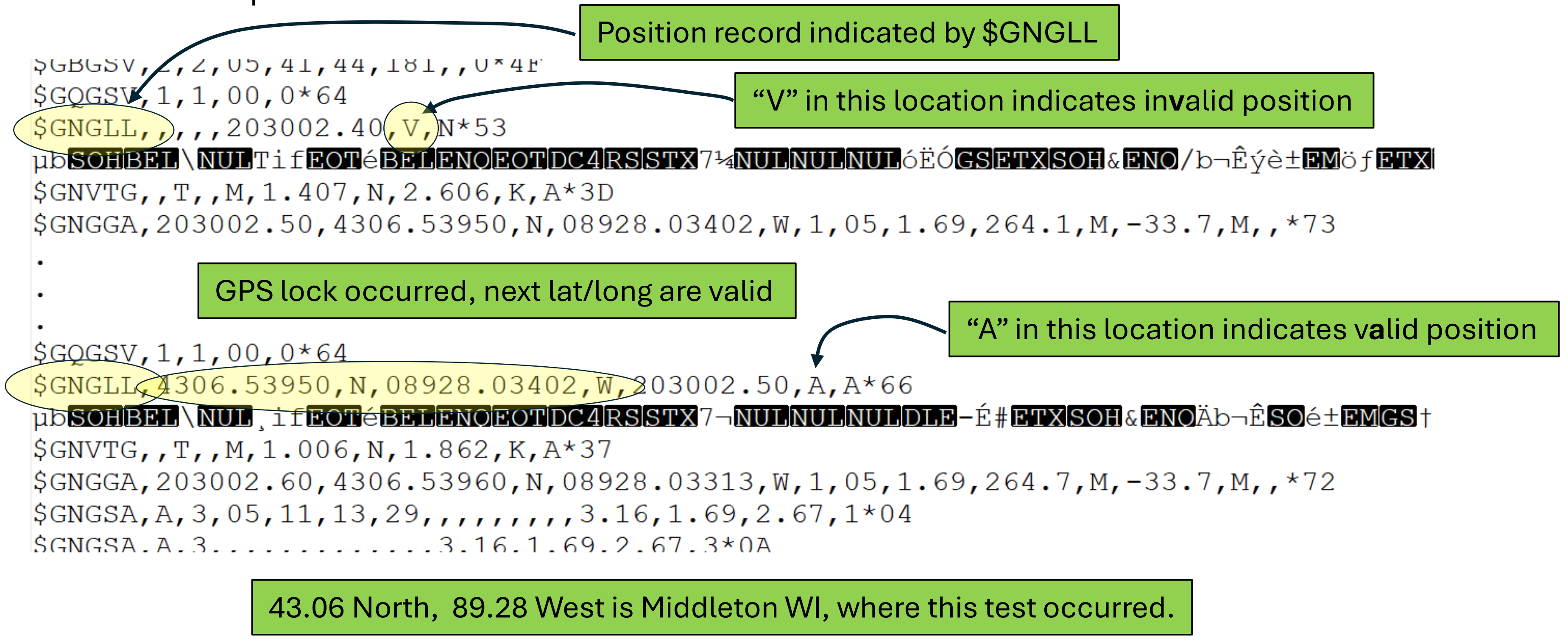
When the Path Controller is "in charge" of the drone it will determine desired pitch and roll of the copter to achieve the desired navigation objectives. There are only two navigation objectives:

1) Hold Current Position
2) Return to Home Position

The Path Controller will have a low PPM watch crystal to enable accurate time keeping *(i.e. incase accurate velocity needs to be derived from position updates)*

Interface GPS Module → Path Controller:

The GPS module “spews” many messages to the Path Controller. The location update rate is 10Hz. Each location update involves many ASCII strings. The Path Controller can ignore most of this. It will parse and look for a single message type (GNPLL) which is position. Shown below are messages coming from the GPS module when position lock first occurs.


Position record indicated by $GNGLL
“V” in this location indicates invalid position
GPS lock occurred, next lat/long are valid
“A” in this location indicates valid position
43.06 North, 89.28 West is Middleton WI, where this test occurred.

```
$GBGSV,2,2,05,41,44,181,,0*4F
$GQGSV,1,1,00,0*64
$GNGLL,,,,,203002.40,V,N*53
µbSOHBEL\NULTifEOTéBELENQEOTDC4RSSTX7¼NULNULNULóËÓGSETXSOH&ENQ/b¬Êýè±EMöƒETX
$GNVTG,,T,,M,1.407,N,2.606,K,A*3D
$GNGGA,203002.50,4306.53950,N,08928.03402,W,1,05,1.69,264.1,M,-33.7,M,,*73
.
.
.
$GQGSV,1,1,00,0*64
$GNGLL,4306.53950,N,08928.03402,W,203002.50,A,A*66
µbSOHBEL\NUL¸ifEOTéBELENQEOTDC4RSSTX7¬NULNULNULDLE-É#ETXSOH&ENQÄb¬ÊSOé±EMGS†
$GNVTG,,T,,M,1.006,N,1.862,K,A*37
$GNGGA,203002.60,4306.53960,N,08928.03313,W,1,05,1.69,264.7,M,-33.7,M,,*72
$GNGSA,A,3,05,11,13,29,,,,,,,,,3.16,1.69,2.67,1*04
$GNGSA,A,3,,,,,,,,,,,,,3.16,1.69,2.67,3*0A
```

When position lock first occurs the Path Controller should send a message to the Flight Controller indicating this.

Interface Flight Controller → Path Controller:

Data packets from Flight Controller to Path Controller will always be 5-bytes in length. The first byte indicates the packet type.

PckType:	Payload Bytes: (4-bytes)	Description:
0x01	{PitchH,PitchL} {RollH,RollL}	The flight controller will always send the desired pitch and roll currently being commanded by the pilot to the Path Controller. The Path Controller will use this along with location info to build a relation.
0x02	{0x00,0x00} {0x00,0x00}	Update in desired yaw. NOTE: Drone does not know its current ordinal heading. Path Controller needs to figure out relationship between pitch/roll and North/West movement. This just lets the Path Controller know this relationship just changed.
0x03	{0x00,0x00} {0x00,0x00}	Begin position hold. Path Controller marks current position and starts sending pitch/roll commands to maintain this position. Should be assumed last pilot commanded pitch/roll were good starting points.
0x04	{0x00,0x00} {0x00,0x00}	Begin return home. Path Controller starts sending pitch/roll commands to Flight Controller to navigate drone back to GPS coordinates where position lock first occurred.
0x05	{0x00,0x00} {0x00,0x00}	End Path controlling. Applies to both position hold and return home. Path Controller stops sending pitch/roll commands. Control returns to pilot.

Interface Path Controller → Flight Controller:

Data packets from Path Controller to Flight Controller will always be 5-bytes in length. The first byte indicates the packet type.

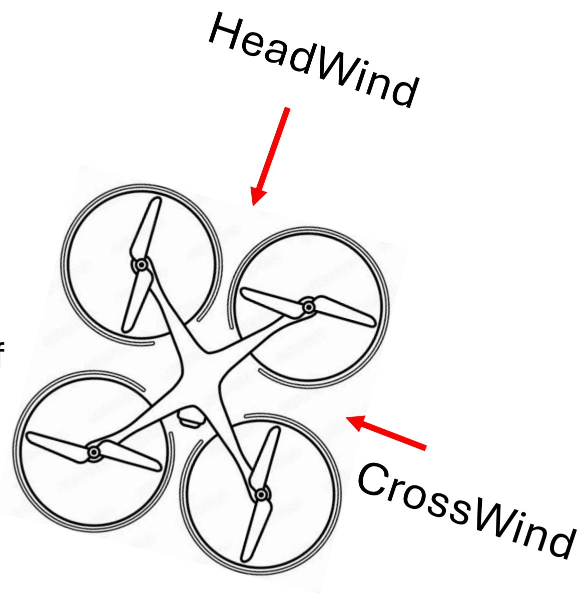
PckType:	Payload Bytes: (4-bytes)	Description:
0x01	{0x00,0x00} {0x00,0x00}	Obtained Lock. This message is sent to inform Flight Controller that position lock has been obtained. Flight Controller will in turn relay this to remote control so pilot is aware.
0x02	{0x00,0x00} {0x00,0x00}	Lock Lost. This message is sent to inform Flight Contoller that position lock has been lost. Flight Controller will in turn relay this to remote control so pilot is aware.
0x03	{PitchH,PitchL} {RollH,RollL}	Desired pitch/roll from Path Controller when performing either position hold or return home. Should always start from last pilot commanded pitch/roll.

Drone Modeling:

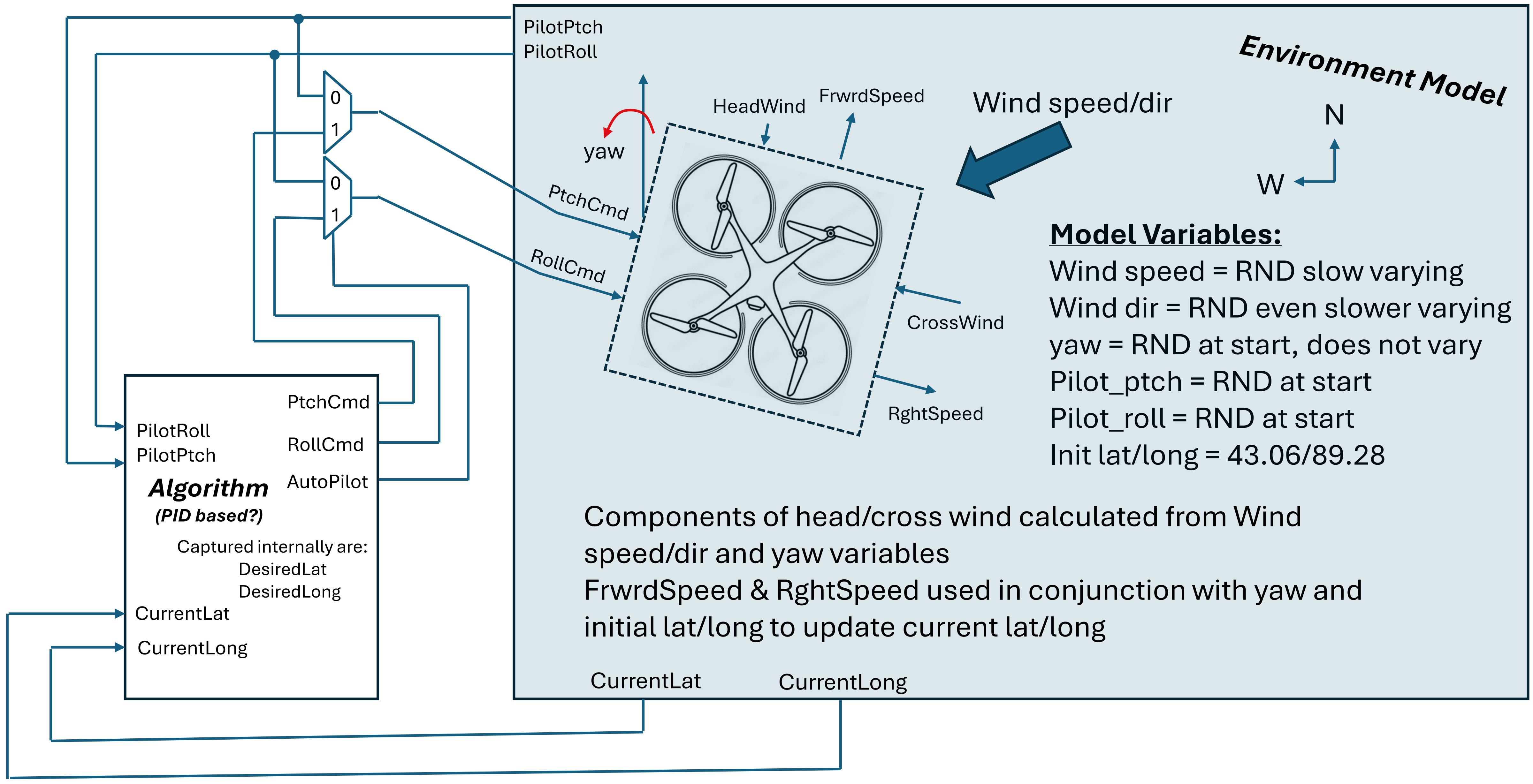
A model of the drones velocity forward and to the left should be developed. It should have internal variables that model acceleration for both forward/backwards and left/right, and derive velocity via integration.

Force forward would be based on –sin(pitch). Resisting wind force would be proportional to square of HeadWind.
Force to right would be based on sin(roll). Resisting wind force would be proportional to square of CrossWind.

It should never be the goal of the Path Controller to have the speed of the drone exceed 900cm/sec



Signal:	Dir:	Description:
HeadWind[11:0]	in	Component of wind (in cm/sec) opposing forward motion of copter. Never exceeding 550cm/sec
CrossWind[11:0]	in	Component of wind (in cm/sec) opposing rightward motion of copter. 550cm/sec max
PitchCmd[10:0]	in	Desired pitch is 1/20 (0.05) degree increments. Pitching left is negative, pitching to go right is positive.
RollCmd[10:0]	in	Desired pitch is 1/20 (0.05) degree increments. Pitching forward is negative, pitching to go backwards is positive. **NOTE:** sign difference between this vs windspeed relative to HeadWind/Pitch.
FrwrdSpeed[11:0]	out	Drones forward speed in cm/sec. This will have to be translated to N/S/E/W speed based on yaw modeling.
RghtSpeed[11:0]	out	Drones speed to left in cm/sec. If drone moving right this would be negative number.



Model Variables:

Wind speed = RND slow varying

Wind dir = RND even slower varying

yaw = RND at start, does not vary

Pilot_ptch = RND at start

Pilot_roll = RND at start

Init lat/long = 43.06/89.28

Components of head/cross wind calculated from Wind speed/dir and yaw variables

FrwrdSpeed & RghtSpeed used in conjunction with yaw and initial lat/long to update current lat/long