**Debugging.py:**

from Debugging.Debug import logToConsole

from Debugging.Debug import logToFile

from Debugging.Debug import logToAll

Where msg is str level is int (lower is more important)

def logToConsole(msg, level)

def logToFile(msg, level)

def logToAll(msg, level)

**Communication Buffer:**

from Communication.CommunicationBuffer import PopCmd

from Communication.CommunicationBuffer import PushCmd

Where inId is a command type enum, inData is a byte array with the data to send

def PushCmd(inID,inData):

def PopCmd()

returns: {cmdID:commandType, data: data}

Where commandType is is a command type enum (COMMAND\_NONE if no command), and data is a byteArray.

**Commands**

from Communication.Commands.Commands import CommandType

class CommandType(Enum):

NO\_COMMAND = 0x00,

ERROR\_STATUS = 0x01,

COMMAND\_QUEUE\_LEN = 0x02,

SYSTEM\_RESET = 0x03,

TEST\_SEQUENCE = 0x04,

LEFT\_MOTOR\_SPEED = 0x11,

RIGHT\_MOTOR\_SPEED = 0x12,

BOTH\_MOTOR\_SPEED = 0x13,

BATTERY\_CURRENT = 0x21,

SYSTEM\_CURRENT = 0x22,

TURRET\_HOR\_ANGLE = 0x31,

TURRET\_VER\_ANGLE = 0x32,

TURRET\_FIRE\_1 = 0x33,

TURRET\_FIRE\_2 = 0x34,

TURRET\_FIRE\_ALL\_1 = 0x35,

TURRET\_FIRE\_ALL\_2 = 0x36,

TURRET\_FIRE\_ALL = 0x37,

TURRET\_LASER\_SET = 0x38,

COMMAND\_INVALID = 0xFF

**Command Encoder:**

from Communication.CommandsEncoder import START\_BYTE

from Communication.CommandsEncoder import END\_BYTE

from Communication.CommandsEncoder import DecodeCmd

from Communication.CommandsEncoder import EncodeCmd

START\_BYTE = bytes([0xA5])

END\_BYTE = bytes([0x5A])

def DecodeCmd(inData):

return {'cmdID':COMMAND\_NONE,'data':byteArray,'valid':1}

Where inId is a command type enum, inData is a byte array with the data to encode

def EncodeCmd(inCmd, inData):