

Common.h
define fundamental data and struct
MATH: PI, RAD_TO_DEG, DEG_TO_RAD
Robot constant: AE_TO_GLOBAL, INVERSE_KINEMATICS_L_, WHEEL_RADIUS, GEAR_RATIO, #define constrain(amt,low,high)
struct position, pointInfo
enum Command

ActionEncoder.h (#include "mbed.h" and #include "common.h")

class ActionEncoder inherited from serial(mbed)

private:
1. encoder global position (defined in common.h)
encoder
2.parameter(wheel_to_center,local_angle,position_angle, timer)
3. data (a union of 24 unit8_t value & 6 float value)
4. some indexes, structs(tempPos, degreePos...) and an 8 char buffer

public:

public.

1. ActionEncoder constructor prototype
2. some useful function(translate, calculatePos, curPosIsAvailable, getR) and a getCurPos function which returns a position struct

ActionEncoder.cpp

ActionEncoder constructor
1. input argument tx,rx pins
2. inherited from class Serial with 115200 baud rate

1. set format 8 bits, no parity bit, 1 stop bit(serial class member function)
2. calculate the uncalculated parameter defined in .h (should be completed in .h)
3. start a timer and check whether it can complete a reading cycle(24 unit8_t) and calculate position within 5 seconds (). Stop the timer afterwards

translate(very important, core function)
1. A finite state machine to read pattern
0x0d, 0x0a, 24-byte-data, 0x0a, 0x0d
2. If complete,
val[0]: position w, in degree, return the delta value
mapped delta value into -180 to 180 and add to w position
Explanation: w range from 0-360, need to know absolute value and how many cycle has been rotated*
Consequencies: may have cumulated error due to low receive rate and
val[3]: position x
val[4]: position y
set flag newDataArrived = true
clear index(a must in every cycle)
(WARNING: serial transmit needs much more time than program execution, it draws down the data refresh rate in our master board
Integrating delta is very dangerous)

