20210408

* The serial buffer is defined in core head file '\\....\Arduino\hardware\Arduino\_STM32\STM32F1\system\libmaple\include\libmaple\usart.h'.
* There may occurs data overflow in MCU receive buffer if high-level control frequency does not match the low-level control frequency as no buffer clear program is written for MCU

20210409

* When high-level controller is stopped with timer object delete in PC high-level controller, a stop command will be sent to MCU for low-level controller, but no guarantee measurement has be made that MCU can correctly receive this command
* Potential measurement-Make sure the main program only stop when info package feedback turn to resume to a safe operation point

20210410

* Caution that before applying the control program, each sensor should be calibrated first
* IMU, load cell, spring stiffness, motor actuation unit parameter, system parameter
* IMU operation algorithm: 6 axis/9 axis
* To adjust sensor feedback item for control, both ADC config and sensor feedback info prcessing program should be adjusted simultaneously to make sure it is obtained correctly with correct command
* ADC enable channel
* ADC detected channel
* IMU operation algorithm (6/9axis)
* Sensor feedback direction (+/- sign is coincides with controller definition or not)
* Sensor feedback processing items in sensorFeedbackPro()
* To increase main-loop running speed, Enabled ADC channel can be adjusted.
* Adjust of ENABLED\_CH & *i* in getADC() and
* Adjust of ENABLED\_CH & *i* & *t* in getADCaverage()
* Before running the program, the motor actuation parameter and rotation direction need to be calibrated
* Motor rotation direction
* Actuation unit parameters:
* Motor current constant, motor driver configuration, gear box
* Pulley radius
* Spring stiffness
* For the Motor enable/disable logic: the motor enable pin is disabled after initialization in MCU program 🡪 will be enabled when the 'mode' flag received from PC indicates normal operation 🡪 will be disable again if ‘mode’ flag indicates stop status
* The stop condition of high-level controller need to be adjusted as practical application required
* For testing: time condition may be enough
* For practical application: At present, the condition can be time condition + mode status condition

20210415

* Yaw angle return to zero logic
* Need to be updated with practical user intention detection strategy
* Timer set in MATLAB Prog
* ‘BusyMode’ : ‘queue’ or ‘drop’
* Frequency adjustment for highest communication frequency
* Serial port set in MATLAB Prog
* Terminator
* BaudRate
* Com port name