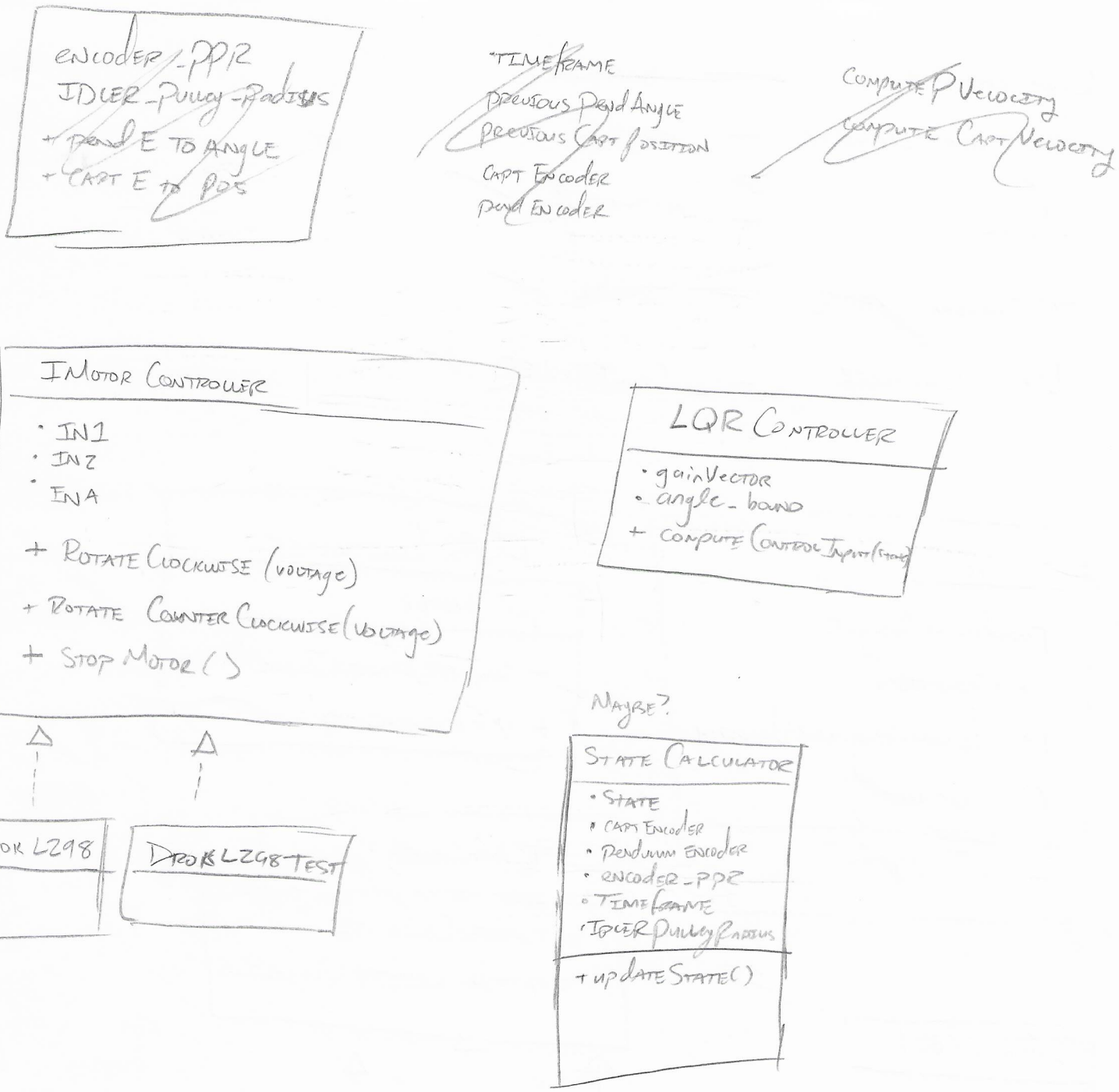

ENCODER_PPR
IDLER_Pulley_Radius
+ PEND E TO ANGLE
+ CART E TO POS
*TIMEFRAME
PREVIOUS PEND ANGLE
PREVIOUS CART POSITION
CART ENCODER
PEND ENCODER
COMPUTE PVELOCITY
COMPUTE CART VELOCITY
IMOTOR CONTROLLER
• IN1
• IN2
• ENA
+ ROTATE CLOCKWISE (VOLTAGE)
+ ROTATE COUNTER CLOCKWISE (VOLTAGE)
+ STOP MOTOR ()
LQR CONTROLLER
• gainVECTOR
• angle_bound
+ COMPUTE CONTROL INPUT(
PROK L298
DROK L298 TEST
MAYBE?
STATE CALCULATOR
• STATE
• CART ENCODER
• PENDULUM ENCODER
• ENCODER_PPR
• TIMEFRAME
• IDLER PULLEY RADIUS
+ UPDATE STATE()

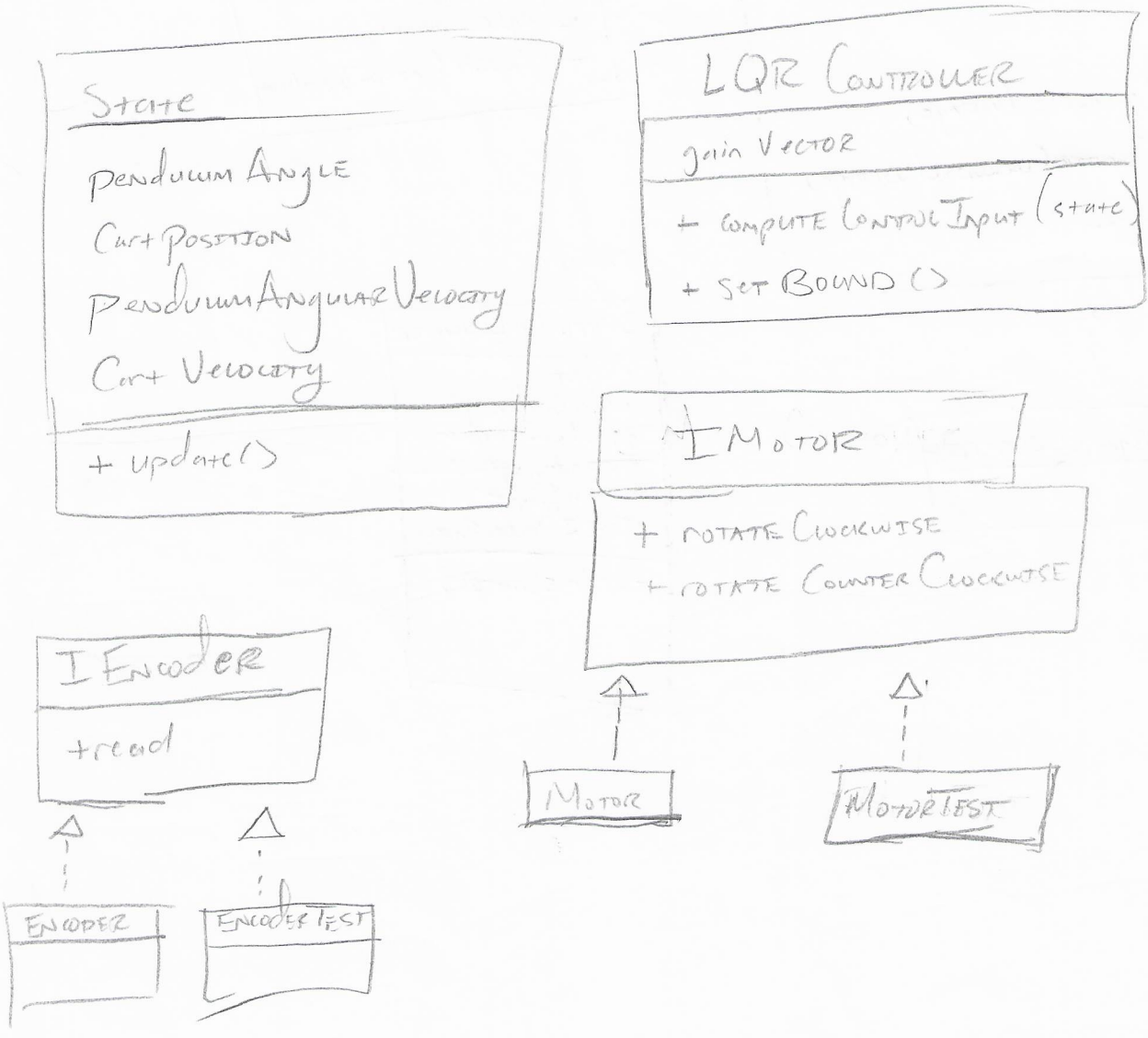
Cart encoder — Sensor

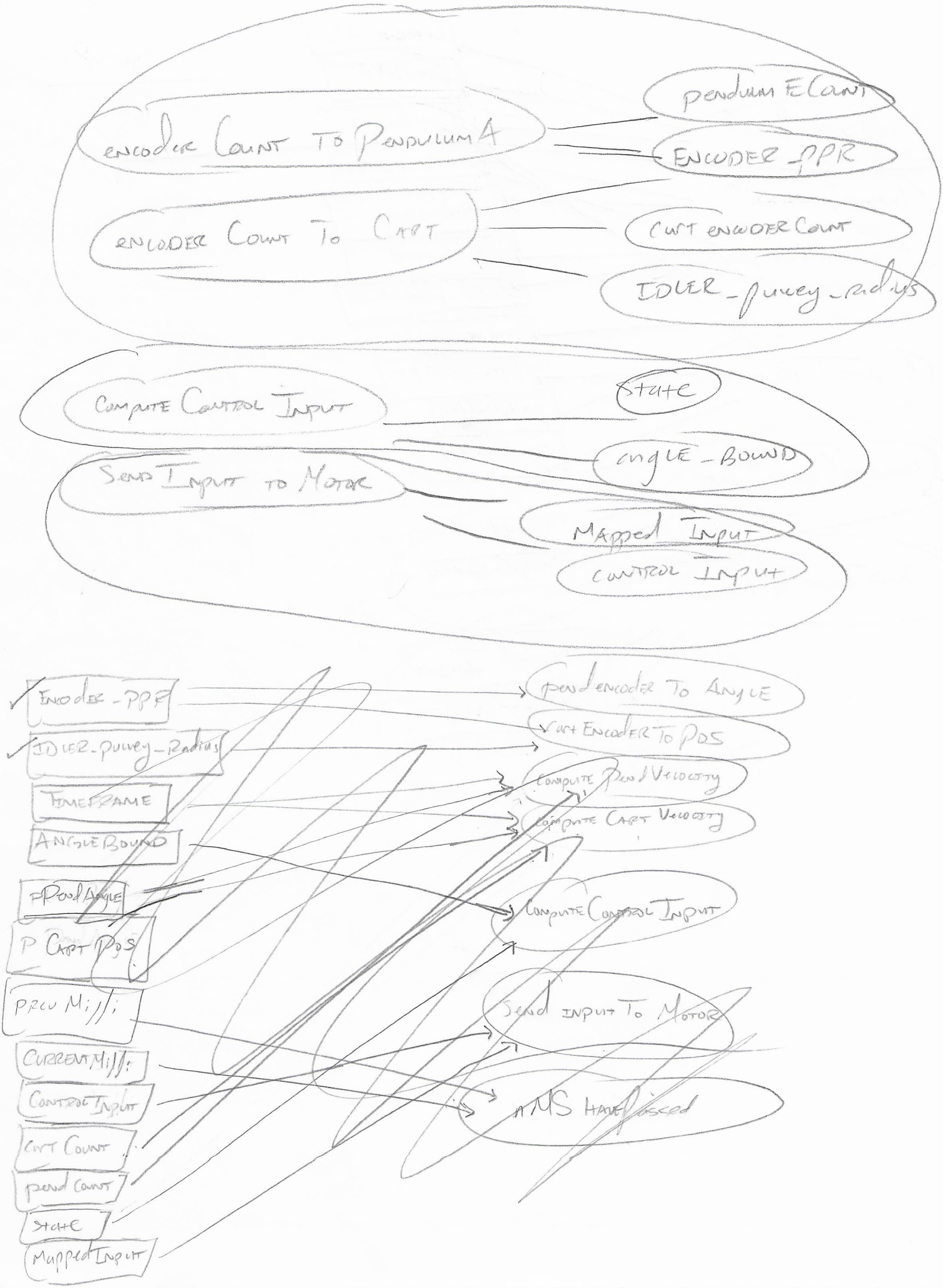
pendulum encoder — Sensor

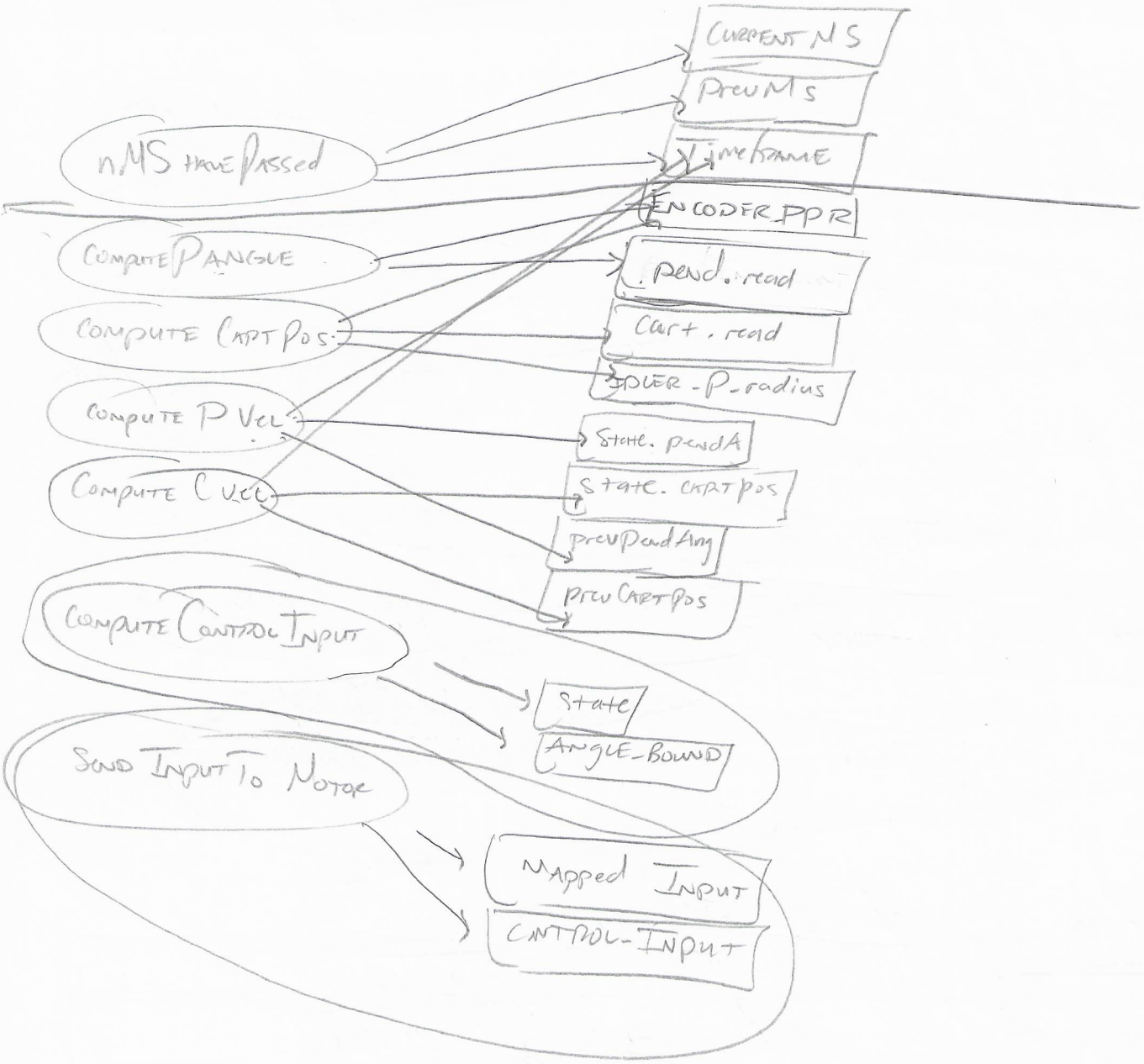
Motor — Actuator ✓

pendulum State — State ✓

LQR controller — Controller ✓




encoder Count To PendulumA
pendulumECount
ENCODER_PPR
encoder Count To Cart
CartEncoderCount
IDLER_pulley_radius
Compute Control Input
State
angle_Bound
Send Input to Motor
Mapped Input
Control Input
ENCODER_PPR
IDLER_pulley_Radius
TIMEFRAME
ANGLEBOUND
pPendAngle
P Cart Pos
prev Mill:
CURRENT Mill:
Control Input
Cart Count
pend Count
State
MappedInput
pendEncoder To Angle
Cart Encoder To Pos
Compute Pend Velocity
Compute Cart Velocity
Compute Control Input
Send Input To Motor
nMS Have Passed


CURRENT_MS
PREV_MS
TIME_FRAME
ENCODER_PPR
pend.read
Cart.read
POLER_P_radius
state.pendA
state.CARTPOS
prevPendAng
PREV CART POS
nMS have Passed
COMPUTE PANGLE
COMPUTE CARTPOS
COMPUTE P VEL
COMPUTE C VEL
COMPUTE CONTROL INPUT
state
ANGLE_BOUND
SEND INPUT TO MOTOR
MAPPED INPUT
CONTROL_INPUT