**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**//Definitions of motor numbers & home sensors**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**M\_verticalMotor = 0 // refL up down**

**M\_linearMotor = 1 // refL, refR linear movement between vials**

**M\_armMotor = 2 // refL move vials from top to bottom**

**M\_pistonMotor = 3 // refL, syringe(RefR) fill syringe**

**M\_headRotateMotor = 4 // refL rotate head top to bottom**

**M\_disposeMotor = 5 // refL push vials out from the machine**

**M\_capHolderMotor = 6 // sensor at In\_capHolderHome. separate motor driver**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**//Definitions of Inputs**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**In\_capHolderHome = 0 // U10**

**In\_pwrDrawer = 1 // U38**

**In\_NeedleDetected = 2 // U11**

**In\_3 = 3 // U36**

**In\_slidingDoor = 4 // U45**

**In\_drawerOverflow = 5 // U15**

**In\_drawerClose = 6 // U14**

**In\_7 = 7 // U25**

**// bottles sensors via relay multiplexers**

**InX\_salineBagIsIn = 0 // U44**

**InX\_vialIsIn1 = 1 // U16**

**InX\_vialIsIn2 = 2 // U46**

**InX\_vialIsIn3 = 3 // U37**

**InX\_vialIsIn4 = 4 // U47**

**InX\_vialIsIn5 = 5 // U48**

**InX\_vialIsIn6 = 6 // U33**

**InX\_mux7 = 7 // U30**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**//Definitions of Outputs**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**Out\_GreenLED = 0 // yellow color when both red and green are on**

**Out\_RedLED = 1 //**

**Out\_PulseCapHolder = 2 // for capHolder motor**

**Out\_VibrateDIR = 3 // outputs DIR for 2 vibrators**

**Out\_CAPHolderDIR\_Down = 4 // 1A output DIR for for capHolderMotor**

**Out\_enaVibrate\_4 = 5 // enable vibrator 4 (bottle 4)**

**Out\_enaVibrate\_56 = 6 // enable vibrator 56 (bottles 56)**

**Out\_Multiplexer = 7 // 1A AND drawer enable**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Defines**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**VIAL\_IS\_LOADED = 1**

**SYRINGE\_IN\_PLACE = 1**

**ANY\_ERROR\_EXISTS = 1**

**WAIT\_TICKS\_200ms = 20 // [200ms]**

**WAIT\_TICKS\_VERY\_SHORT = 100 // [ 1s]**

**WAIT\_TICKS\_2S = 200 // [ 2s]**

**WAIT\_TICKS\_4S = 400 // [ 4s]**

**WAIT\_TICKS\_DISPOSE = 300 // [ 3s]**

**WAIT\_TICKS\_ARM = 2000 // [20s]**

**WAIT\_TICKS = 1000 // [10s]**

**WAIT\_TICKS\_LINEAR = 1500 // [15s]**

**WAIT\_TICKS\_LONG = 1500 // [15s]**

**WAIT\_TICKS\_FILL\_VIAL = 4000 // [40s]**

**WAIT\_TICKS\_DRAW = 4000 // [40s]**

**LED\_ON = 1**

**LED\_OFF = 0**

**DEF\_DRAW\_WAIT\_TIME\_25M\_22G = 200 // default for 25 mm needle length and 22 Gauge = 2[s]**

**DEFAULT\_NEEDLE\_LENGTH = 25**

**DEFAULT\_DRAW\_VOLUME = 1000 // 1[ml]**

**CAP\_SENSOR\_BLOCKED = 0**

**MINIMUM\_DRAW\_VOLUME = 500 // not less than 0.1 ml**

**LAST\_VIAL = 6**

**FIRST\_VIAL\_FILL = 4**

**FIRST\_VIAL\_VIBRATE = 4**

**OVERFLOW = 0 // the sesor sees vials in the drawer**

**DRAWER\_IS\_CLOSED = 1**

**DOOR\_IS\_CLOSED = 1**

**//CM**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// User variables map (bank 2)**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// prameters from Trinamic to PC:**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// GB\_currentVersion = 0 // Current SW version // moved to GB\_222**

**GB\_0 = 0**

**GB\_Syringe\_Type = 1**

**GB\_2 = 2 // = 100 = 1[s]**

**GB\_3 = 3 // GB\_InitDone = 3 // INIT\_MOTORS (func 0) done. starts automaticaly**

**GB\_4 = 4 //**

**GB\_5 = 5 //**

**GB\_RobotSerialNumber = 6 // (2030006) 2 = CM, 03 = HW ver, 0006 = machine #) // Write serial number -> will be set by calibration on GB\_6**

**GB\_7 = 7**

**// the following 2 global parameters will appear at motor 0 (vertical)**

**// GB\_verticalCapPos = 8 // syringe loading**

**// GB\_BumpPosVert = 9 // needle touches the vial 1 rubber at top ~ -8792**

**GB\_needleLength = 10 // 16, 25, 40 [mm]**

**GB\_needleGauge = 11 // 14, 16, 18, 21, 22, 23 [gauge]**

**GB\_cyclesTotal = 12 //**

**// GB\_setBumpBottom = 13 // needle touches the vial 4 rubber at bottom**

**GB\_14 = 14 //**

**GB\_15 = 15 //**

**GB\_16 = 16 //**

**GB\_17 = 17 //**

**GB\_18 = 18 //**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// prameters from PC to Trinamic:**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_19 = 19**

**GB\_ShowOverride = 20 // 1 = will not dispose vials and also loop DRAW\_DOSE function // ??? (GP 28 error)**

**GB\_21 = 21**

**GB\_disposeYN = 22 //**

**// GB\_DisposeDropVialsPos = 23 //**

**// GB\_DisposeDropVials456Pos = 24 // arm at disposal position for vials 4 5 6**

**GB\_25 = 25**

**GB\_26 = 26**

**GB\_27 = 27**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// CM Errors**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_errors\_syringe\_bag = 28**

**// bit errors for parameter 28**

**BitEr\_bagIsMissing = %00000000001 // bit 1**

**BitEr\_syringePoppedOut = %00000000010 // bit 2 syringe popped out during cycle**

**BitEr\_volumeExceedsBag = %00000000100 // bit 4**

**BitEr\_SyringeIsIn = %00000001000 // bit 8 FIND\_HOMES error or INIT\_DRAW\_DOSE, Syringe is in the system**

**BitEr\_SyringeMissing = %00000010000 // bit 16 DRAW\_DOSE error, missing the syringe**

**BitEr\_machineAborted = %00000100000 // bit 32**

**BitEr\_g7 = %00001000000 // bit 64**

**BitEr\_g8 = %00010000000 // bit 128**

**BitEr\_g9 = %00100000000 // bit 256**

**GB\_any\_Error = 29 // any error occured**

**GB\_errors\_M\_verticalMotor = 30**

**GB\_errors\_M\_linearMotor = 31**

**GB\_errors\_M\_armMotor = 32**

**GB\_errors\_M\_pistonMotor = 33**

**GB\_errors\_M\_headRotateMotor = 34**

**GB\_errors\_M\_disposeMotor = 35**

**GB\_errors\_M\_capHolderMotor = 36**

**// motor errors for parameters 30-36 Motors**

**BitEr\_leftRefSensor = %00000001 // bit 1 left ref sensor**

**BitEr\_rightRefSensor = %00000010 // bit 2 right ref sensor**

**BitEr\_homeNotFound = %00000100 // bit 4 did not find home**

**BitEr\_TimeOut = %00001000 // bit 8**

**// BitEr\_calibrationErr = %00010000 // bit 16**

**BitEr\_m6 = %00100000 // bit 32**

**BitEr\_m7 = %01000000 // bit 64**

**BitEr\_m8 = %10000000 // bit 128**

**GB\_errors\_Vial\_1 = 37**

**GB\_errors\_Vial\_2 = 38**

**GB\_errors\_Vial\_3 = 39**

**GB\_errors\_Vial\_4 = 40**

**GB\_errors\_Vial\_5 = 41**

**GB\_errors\_Vial\_6 = 42**

**// bit errors for parameter 37-42 Vials**

**BitEr\_VialTooSmall = %00000000001 // bit 1**

**BitEr\_VialMissing = %00000000010 // bit 2**

**BitEr\_VialPoppedOut = %00000000100 // bit 4 vial popped out during cycle**

**BitEr\_v4 = %00000001000 // bit 8**

**BitEr\_v5 = %00000010000 // bit 16**

**BitEr\_v6 = %00000100000 // bit 32**

**BitEr\_v7 = %00001000000 // bit 64**

**BitEr\_v8 = %00010000000 // bit 128**

**GB\_errors\_findHome = 43**

**// bit errors for parameter 43**

**BitEr\_syringeIsInwhileFindHome = %00000001 // bit 1**

**BitEr\_h2 = %00000010 // bit 2**

**BitEr\_capHolderIsInWhileFindHome = %00000100 // bit 4**

**BitEr\_h4 = %00001000 // bit 8**

**BitEr\_h5 = %00010000 // bit 16**

**BitEr\_h6 = %00100000 // bit 32**

**BitEr\_h7 = %01000000 // bit 64**

**BitEr\_h8 = %10000000 // bit 128**

**GB\_errors\_wrong\_PC\_command = 44**

**// bit errors for parameter 44**

**BitEr\_expecting\_GP5\_10\_OR\_30 = %00000001 // bit 1**

**BitEr\_expecting\_WAITING\_DISPENSE = %00000010 // bit 2**

**BitEr\_c3 = %00000100 // bit 4**

**BitEr\_vibrateParemeterError = %00001000 // bit 8**

**BitEr\_c5 = %00010000 // bit 16**

**BitEr\_c6 = %00100000 // bit 32**

**BitEr\_c7 = %01000000 // bit 64**

**BitEr\_c8 = %10000000 // bit 128**

**GB\_special\_Error = 45**

**// motor errors parameter 45**

**BitEr\_slidingDoorIsOpen = %00000001 // bit 1**

**BitEr\_drawerOverflow = %00000010 // bit 2**

**BitEr\_No\_vials = %00000100 // bit 4 DRAW\_DOSE error, missing the vial**

**BitEr\_drawerIsOpen = %00001000 // bit 8**

**BitEr\_vialNotDefined = %00010000 // bit 16**

**BitEr\_s5 = %00100000 // bit 32**

**BitEr\_s6 = %01000000 // bit 64**

**BitEr\_s7 = %10000000 // bit 128**

**// the following global parameters will appear at their motor**

**// GB\_RecapPositionlinear = 46 // to hold the cap for recap**

**// GB\_ArmVialPosition = 47 // cap holder is under the syringe**

**// GB\_PistonHomePos = 48 // for empty syringe**

**// GB\_HeadRotateHomePos = 49 // center of syringe cap**

**// GB-50 = 50 // at the top**

**// GB\_SetVial4BottomLinear = 51 // linear motor**

**// GB\_headAtBottom = 52**

**// GB\_linearCenterOfVial1 = 53**

**// GB\_linearLoading = 54 //// GB\_CapLimitSwitchDisable = 54 /// ??? //// L.E.D ////**

**// GB\_armAtBottom = 55 //**

**// 55 is the last GB that can also be stored permanently in the EEPROM**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// motor parameters**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Motor\_0 = M\_verticalMotor**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_verticalCapPos = 8 // syringe loading 380**

**GB\_BumpPosVert = 9 // needle touches the vial 1 rubber at top ~ -8792**

**GB\_setBumpBottom = 13 // needle touches the vial 4 rubber at bottom**

**GB\_needleLengthInMicroSteps = 60 // verticalMicroStepsPerMM \* GB\_needleLength = 5039 [steps] for 40 [mm]**

**GB\_Calced\_Top\_Vert\_Needle\_Down\_Allowed = 61 // GB\_61 = 61**

**GB\_62 = 62**

**GB\_63 = 63**

**GB\_64 = 64**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Motor\_1 = M\_linearMotor**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_SetVial4BottomLinear = 51**

**GB\_linearCenterOfVial1 = 53 //**

**GB\_linearLoading = 54 //**

**GB\_currentVialLinearLocation = 65**

**GB\_LinearCenterOfBag = 66**

**GB\_67 = 67**

**GB\_68 = 68**

**GB\_69 = 69**

**// ++++++++++++++++++++++++++++++++**

**// Motor\_2 = M\_armMotor**

**// ++++++++++++++++++++++++++++++++**

**GB\_DisposeDropVials456Pos = 24 // arm at disposal position for vials 4 5 6**

**GB\_ArmVialPosition = 47 // arm holder is above the syringe -460**

**GB\_armAtBottom = 55 // 15546 Arm at bottom just below syring**

**GB\_armMicroStepsPerMM = 70 // 31**

**//GB\_limitarmBentMicroS = 71**

**//GB\_NeedleArmError = 72 // error of needle at calholder position**

**GB\_73 = 73**

**GB\_74 = 74**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Motor\_3 = M\_pistonMotor**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_PistonHomePos = 48 // for empty syringe 345 [0] => need to redesign - adujstment to new syringe holder and diferent syringes**

**// GB\_microL\_per\_100mm\_current = 161 // 6522**

**GB\_DrawWaitTime = 75**

**GB\_pistonMicroStepPer100microL = 76 // 10000 \* pistonMicroStepsPerMM / GB\_microL\_per\_100mm\_current = 224**

**GB\_77 = 77 //**

**GB\_78 = 78**

**GB\_79 = 79**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Motor\_4 = M\_headRotateMotor**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_HeadRotateHomePos = 49 // center of cap 16232 // 16240**

**GB\_headAtBottom = 52**

**GB\_80 = 80**

**GB\_rotateMicroStepsPerMM = 81**

**//GB\_limitNeedleBentMicroS = 82**

**GB\_NeedleVialError = 83 // for vial center calculation**

**GB\_84 = 84**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Motor\_5 = M\_disposeMotor**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_51 = 51 // not used**

**GB\_DisposeDropVialsPos = 23**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// Motor\_6 = M\_capHolderMotor**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// GB\_CapLimitSwitchDisable = 54 // was erased once //// L.E.D ////**

**GB\_CapHolderHolds = 86 // "1" = holds, "0" = open**

**GB\_CapHolderPulses = 87**

**GB\_slowCapHolder = 88**

**GB\_capWaitLoops = 89**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// parameters for CM**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// microLtoWithdraw values**

**GB\_microLtoWithdraw\_current = 90**

**GB\_microLtoWithdraw\_1 = 91**

**GB\_microLtoWithdraw\_2 = 92**

**GB\_microLtoWithdraw\_3 = 93**

**GB\_microLtoWithdraw\_4 = 94**

**GB\_microLtoWithdraw\_5 = 95**

**GB\_microLtoWithdraw\_6 = 96**

**// TODO**

**//GB\_currentVolume\_microL\_1 = TBD**

**//GB\_currentVolume\_microL\_2 = TBD**

**//GB\_currentVolume\_microL\_3 = TBD**

**//GB\_currentVolume\_microL\_4 = TBD**

**//GB\_currentVolume\_microL\_5 = TBD**

**//GB\_currentVolume\_microL\_6 = TBD**

**GB\_CmdInProcess = 97**

**// SW states**

**GB\_CurrentState = 98 // defines the current state of the process**

**WAITING\_INIT\_CM = 10 // this is the state after the power up as well**

**RUNNING\_INIT\_CM = 20**

**WAITING\_DISPENSE = 30 // at this state it is possible as well to run INIT\_DRAW\_DOSE (6)**

**RUNNING\_DISPENSE = 40**

**STATE\_50 = 50**

**STATE\_60 = 60**

**STOPPED\_ON\_ERROR = 70**

**STOPPED\_TIME\_OUT = 80**

**ABORTED = 90**

**GB\_microLinBAG = 99**

**GB\_moveManualBackwards = 100 // 1 = forward; -1 = backward**

**// vialSize\_microL values**

**GB\_vialSize\_microL\_1 = 101**

**GB\_vialSize\_microL\_2 = 102**

**GB\_vialSize\_microL\_3 = 103**

**GB\_vialSize\_microL\_4 = 104**

**GB\_vialSize\_microL\_5 = 105**

**GB\_vialSize\_microL\_6 = 106**

**GB\_vialSize\_microL\_current = 107**

**GB\_BagSize\_microL = 108**

**GB\_109 = 109**

**GB\_110 = 110**

**GB\_rotateVialsDown = 111**

**Bit\_vibrate0 = %00000001 // bit 1**

**Bit\_vibrate1 = %00000010 // bit 2**

**Bit\_vibrate2 = %00000100 // bit 4**

**Bit\_vibrate3 = %00001000 // bit 8**

**Bit\_vibrate4 = %00010000 // bit 16**

**Bit\_vibrate5 = %00100000 // bit 32**

**GB\_vibration4IsNeeded = 112**

**GB\_vibration56IsNeeded = 113**

**// vibration time for vials [seconds]**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_vibrationTime\_4 = 114**

**GB\_vibrationTime\_4\_calc = 115**

**GB\_vibrationTime\_56 = 116**

**GB\_vibrationTime\_56\_calc = 117**

**GB\_vibrationLocation = 118**

**Bit\_vibrateUp = %00000001 // bit 1**

**Bit\_vibrateDown = %00000010 // bit 2**

**Bit\_vibrateHalfway = %00000100 // bit 4**

**Bit\_vibrateLocation3 = %00001000 // bit 8**

**Bit\_vibrateLocation4 = %00010000 // bit 16**

**Bit\_vibrateLocation5 = %00100000 // bit 32**

**// vibration strength for vials 4,56 - 1 / 2 / 3 / 4 / 5**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_vibrStrengthPercentCalc = 119 // set up %**

**GB\_PwmDutyCycleMS = 120 // calculated [ms]**

**GB\_vibrator4done = 121**

**GB\_vibrator56done = 122**

**GB\_vibrationDutyCyclePercent = 123 // 10-100 [%]**

**// vibration cycle time for vials 4,56 - [ms]**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_vibrationCycleMS = 124 // calculation to ms**

**GB\_vibrationHz = 125 // data input**

**GB\_126 = 126**

**GB\_vibrationIsNeeded = 127**

**GB\_128 = 128**

**// microLtoFill values**

**GB\_microLtoFill\_current = 130**

**GB\_microLtoFill\_1 = 131 //// L.E.D //// for possible future use**

**GB\_microLtoFill\_2 = 132 //// L.E.D //// for possible future use**

**GB\_microLtoFill\_3 = 133 //// L.E.D //// for possible future use**

**GB\_microLtoFill\_4 = 134**

**GB\_microLtoFill\_5 = 135**

**GB\_microLtoFill\_6 = 136**

**//GB\_microLtoFill\_1 = 131 //// L.E.D //// for possible future use**

**//GB\_microLtoFill\_2 = 132 //// L.E.D //// for possible future use**

**//GB\_microLtoFill\_3 = 133 //// L.E.D //// for possible future use**

**//GB\_microLtoFill\_4 = 134**

**//GB\_microLtoFill\_5 = 135**

**//GB\_microLtoFill\_6 = 136**

**GB\_FillingIsNeeded = 140**

**GB\_Filling1IsNeeded = 141 //// L.E.D //// for possible future use**

**GB\_Filling2IsNeeded = 142 //// L.E.D //// for possible future use**

**GB\_Filling3IsNeeded = 143 //// L.E.D //// for possible future use**

**GB\_Filling4IsNeeded = 144**

**GB\_Filling5IsNeeded = 145**

**GB\_Filling6IsNeeded = 146**

**// current syringe**

**GB\_Max\_Volume\_current = 160**

**GB\_microL\_per\_100mm\_current = 161**

**GB\_Syring\_Length\_current = 162**

**GB\_readyToFill = 163**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_GoToAbsoluteInSteps = 182**

**// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_numberOfKicks = 183 // 10**

**GB\_rotateBubblesUM = 184 // 750 = 0.75mm**

**GB\_pistonBubblesPullMicroL = 185 // 300 \* microL = 0.3ml**

**GB\_pistonBubblesPushMicroL = 186 // 150 \* microL = 0.3ml**

**GB\_dispenseCycleTime01s = 187**

**GB\_dispenseCycleTimeMS = 188**

**GB\_PigWasReplaced = 190 //**

**GB\_inHomeCapHolderMotor = 191 // signals when to check 3 times for syringe poped out**

**GB\_MulCenterOfVial = 192 // for running average calculations**

**GB\_adjustmentsTotal = 193**

**GB\_TouchedLeftRef = 195**

**GB\_dipperInterruptHight = 196**

**GB\_microLinVial\_1 = 197**

**GB\_microLinVial\_2 = 198**

**GB\_microLinVial\_3 = 199**

**GB\_microLinVial\_4 = 200**

**GB\_microLinVial\_5 = 201**

**GB\_microLinVial\_6 = 202**

**GB\_microLinVial\_current = 203**

**GB\_current\_Vial = 204**

**// the following definitions will be loaded on the bit definitions GB**

**// when checkin vial existance, these definitions will be used as AND on the GB\_vialsExist parameter**

**GB\_vial1Bit = 211 // VIAL1\_BIT**

**GB\_vial2Bit = 212 // VIAL2\_BIT**

**GB\_vial3Bit = 213 // VIAL3\_BIT**

**GB\_vial4Bit = 214 // VIAL4\_BIT**

**GB\_vial5Bit = 215 // VIAL5\_BIT**

**GB\_vial6Bit = 216 // VIAL6\_BIT**

**GB\_BagBit = 217 // BAG\_BIT**

**GB\_vialsExist = 218**

**Bit\_vial1 = %00000001 // bit 1**

**Bit\_vial2 = %00000010 // bit 2**

**Bit\_vial3 = %00000100 // bit 4**

**Bit\_vial4 = %00001000 // bit 8**

**Bit\_vial5 = %00010000 // bit 16**

**Bit\_vial6 = %00100000 // bit 32**

**Bit\_bag = %01000000 // bit 64**

**GB\_currentVersion = 222 // Current SW version**

**GB\_readyToDraw = 224 // after verifyReady**

**GB\_initialVolume = 225 // in syringe**

**GB\_MotorIsMoving = 226 //**

**GB\_HomingDone = 228 //**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// prameters from PC to Trinamic:**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**GB\_UnitsToMoveManual = 229 // for manual movement, put here the microMeter or Mili Degree to move (+/-)**

**GB\_TempNumHolder = 230**

**GB\_airToPullBefore = 231 // 20 in Percentage. Should be reduced when vial is full**

**GB\_232 = 232**

**GB\_233 = 232**

**GB\_234 = 232**

**GB\_motorNumForHome = 235 // 0=vertical, 1=linear, 2=arm, 3=piston, 4=rotate, 5=dispose, 6=capHolder**

**GB\_236 = 236 //**

**GB\_microLbagToFill = 237 // microL to withdraw in BAG**

**GB\_skipCheckVial456 = 250**

**GB\_skipCheckBag = 251**

**GB\_foundCenterOfNeedle = 252**

**GB\_DrawFromVialWaitTime = 253 //// L.E.D //// - not sure i still want these ... //// L.E.D ////**

**GB\_vialsToDrawFromCounter = 254 // temporary variable //// L.E.D ////**

**GB\_InterruptCount = 255 // temorary variable 5**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// CODE STARTS HERE**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**// subroutines initialize from PC // "starting address":**

**FUNC\_0: CSUB FIRST\_RUN //v will run on power ON**

**JA backGroundProcess**

**FUNC\_2: CSUB Draw\_654321 // for testing**

**JA backGroundProcess**

**FUNC\_4: CSUB Eyals\_dummy\_sub**

**JA backGroundProcess**

**FUNC\_6: JA INIT\_CM // Run the initial setup, detect if there is a initial volume in the system**

**JA backGroundProcess**

**FUNC\_8: CSUB DRAW\_DOSE // Draw the requested activity**

**JA backGroundProcess**

**FUNC\_10: CSUB dropVials**

**JA backGroundProcess**

**func\_12: CSUB ABORT // stop all motors and program**

**JA backGroundProcess**

**func\_14: CSUB VERIFY\_READY\_DRAW // checks if the vial' syrine ready to draw**

**JA backGroundProcess**

**FUNC\_16: CSUB INIT\_MOTORS // Initialize parameters.**

**JA backGroundProcess**

**func\_18: CSUB FIND\_HOMES**

**JA backGroundProcess**

**func\_20: CSUB HomeCalibration**

**JA backGroundProcess**

**func\_22: CSUB LEDS\_OFF**

**JA backGroundProcess**

**func\_24: CSUB RED\_ON**

**JA backGroundProcess**

**func\_26: CSUB GREEN\_ON**

**JA backGroundProcess**

**func\_28: CSUB YELLOW\_ON**

**JA backGroundProcess**

**func\_30: CSUB positionVerticalMotor // will not search bottom side. for debugging**

**JA backGroundProcess**

**func\_32: CSUB initInterrups**

**JA backGroundProcess**

**func\_34: CSUB verifyVIAL**

**JA backGroundProcess**

**func\_36: CSUB screenAllVials**

**JA backGroundProcess**

**func\_38: CSUB PositionHeadRotateMotor**

**JA backGroundProcess**

**func\_40: CSUB verticalMotorTOerr**

**JA backGroundProcess**

**func\_42: CSUB homeLinearMotor**

**JA backGroundProcess**

**func\_44: CSUB capHolderMotorTOerr**

**JA backGroundProcess**

**func\_46: CSUB pistonMotorTOerr**

**JA backGroundProcess**

**func\_48: CSUB headRotateMotorTOerr**

**JA backGroundProcess**

**func\_50: CSUB homeArmMotor**

**JA backGroundProcess**

**func\_52: CSUB HomeDisposeMotor**

**JA backGroundProcess**

**func\_54: CSUB incrementCycles**

**JA backGroundProcess**

**func\_56: CSUB positionLinearMotor**

**JA backGroundProcess**

**func\_58: CSUB homeHeadRotateMotor**

**JA backGroundProcess**

**// for FUNC\_6 (INIT\_CM)**

**func\_60: CSUB checkSyringeSensor**

**JA backGroundProcess**

**func\_62: CSUB checkNoSyringe**

**JA backGroundProcess**

**func\_64: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_66: CSUB checkSyrPoppedOut**

**JA backGroundProcess**

**// for FUNC\_8 (DRAW\_DOSE)**

**func\_68: CSUB moveAboveVial // func\_68: CSUB moveAboveVialFromCenter**

**JA backGroundProcess**

**func\_70: CSUB dummy\_sub // checkDrawVolume**

**JA backGroundProcess**

**func\_72: CSUB startPullAir70**

**JA backGroundProcess**

**func\_74: CSUB deCap**

**JA backGroundProcess**

**func\_76: CSUB moveBelowVial**

**JA backGroundProcess**

**func\_78: CSUB insertNeedleIntoVial**

**JA backGroundProcess**

**func\_80: CSUB push70air**

**JA backGroundProcess**

**func\_82: CSUB drawVial**

**JA backGroundProcess**

**func\_84: CSUB ClearRunningErrors**

**JA backGroundProcess**

**func\_86: CSUB moveSlowlyBottom**

**JA backGroundProcess**

**func\_88: CSUB bumpPlunger**

**JA backGroundProcess**

**func\_90: CSUB recapSyringe**

**JA backGroundProcess**

**func\_92: CSUB startHomeDisposeMotor**

**JA backGroundProcess**

**func\_94: CSUB insertNeedleIntoVialBelow**

**JA backGroundProcess**

**func\_96: CSUB positionArmMotor**

**JA backGroundProcess**

**func\_98: CSUB setTestParams**

**JA backGroundProcess**

**func\_100: CSUB drawVialMoreBack**

**JA backGroundProcess**

**func\_102: CSUB homeVerticalMotor**

**JA backGroundProcess**

**func\_104: CSUB homePistonMotor**

**JA backGroundProcess**

**func\_106: CSUB positionPistonMotor**

**JA backGroundProcess**

**func\_108: CSUB ejectSyringeFromTopVial**

**JA backGroundProcess**

**func\_110: CSUB ejectSyringeFromBottomVial**

**JA backGroundProcess**

**func\_112: CSUB dummy\_sub**

**JA backGroundProcess**

**// more functions**

**func\_114: CSUB Vibrate**

**JA backGroundProcess**

**func\_116: CSUB checkVialPoppedOut**

**JA backGroundProcess**

**func\_118: CSUB holdCap**

**JA backGroundProcess**

**func\_120: CSUB stopVibrate**

**JA backGroundProcess**

**func\_122: CSUB homeCapHolderMotor**

**JA backGroundProcess**

**func\_124: CSUB startHomePistonMotor**

**JA backGroundProcess**

**func\_126: CSUB defaultVibrate**

**JA backGroundProcess**

**func\_128: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_130: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_132: CSUB dummy\_sub**

**JA backGroundProcess**

**// manual movements functions**

**func\_134: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_136: CSUB VerticalManual**

**JA backGroundProcess**

**func\_138: CSUB LinearMotorManual**

**JA backGroundProcess**

**func\_140: CSUB armMotorManual**

**JA backGroundProcess**

**func\_142: CSUB PistonManual**

**JA backGroundProcess**

**func\_144: CSUB RotationManual**

**JA backGroundProcess**

**func\_146: CSUB VerticalManualAbsolute**

**JA backGroundProcess**

**func\_148: CSUB LinearMotorManualAbsolute**

**JA backGroundProcess**

**func\_150: CSUB armMotorManualAbsolute**

**JA backGroundProcess**

**func\_152: CSUB PistonManualAbsolute**

**JA backGroundProcess**

**func\_154: CSUB RotationManualAbsolute**

**JA backGroundProcess**

**func\_156: CSUB DisposeManualAbsolute**

**JA backGroundProcess**

**func\_158: CSUB DisposeManual**

**JA backGroundProcess**

**func\_160: CSUB CapHolderManual**

**JA backGroundProcess**

**func\_162: CSUB testCapHolder**

**JA backGroundProcess**

**func\_164: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_166: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_168: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_170: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_172: CSUB dummy\_sub**

**JA backGroundProcess**

**func\_174: CSUB dummy\_sub**

**JA backGroundProcess**

**//++++++++++++++++++++++++++++++++++++++++++++++++++++++++**