```
5077 //*****************************************************************************
5078 //**  SETUP  ******************************************************************
5079 //***********  SETUP  *********************************************************
5080 //*******************  SETUP  ***************************************************
5081 //*****************************  SETUP  ****************************************
5082 //***********************************  SETUP  *********************************
5083 //*****************************************************************************
5084 //
5085 //  Name: setup
5086 //
5087 //  Modification date:
5088 //  Changed by:
5089 //
5090 //  Function:
5091 //  Alle actions for initializing the FSM are processed.
5092 //  - Initialize the serial monitor channel
5093 //  - Initialize the communication channel to HMI (when used)
5094 //  - Initialize the watchdog LED (pin13)
5095 //  - Set up the timer pointer list.
5096 //  - Check the configuration of the defined I/O tags
5097 //  - Initialize the RTC
5098 //  - Initialize all defined I/O
5099 //  - Define the initial state of the FSM; START
5100 //  - Send the GMI messages to HMI (when applied)
5101 //
5102 //*****************************************************************************
5103 void setup()  {
5104
5105
5106   int i;
5107   time_t pctime;
5108   DateTime Now;
5109   int IntYear;
5110   int IntMonth;
5111   int IntDay;
5112   int IntHour;
5113   int IntMinute;
5114   int IntSecond;
5115
5116   Serial.begin(9600);
5117   Serial.println(F("Arduino 2560 "));
5118   Serial.println(F("Program made by Siemonsma"));
5119   Serial.println("");
5120   Serial.println(F("Standard v40, 160516.001"));
5121   Serial.println("");
5122   Serial.println("");
5123
5124   if (UseHMISerial)
5125     {
5126       pinMode(19,INPUT_PULLUP);
5127       Serial1.begin(115200);
5128       //Serial1.begin(9600);
5129   }
5130
5131   // For the GPS
5132   pinMode(17, INPUT_PULLUP);
```

```
5133    Serial2.begin(9600);
5134    Serial2.println(F("$PUBX,40,RMC,0,0,0,0*47")); //RMC OFF
5135    Serial2.println(F("$PUBX,40,VTG,0,0,0,0*5E")); //VTG OFF
5136    Serial2.println(F("$PUBX,40,GGA,0,0,0,0*5A")); //CGA OFF
5137    Serial2.println(F("$PUBX,40,GSA,0,0,0,0*4E")); //GSA OFF
5138    Serial2.println(F("$PUBX,40,GSV,0,0,0,0*59")); //GSV OFF
5139    Serial2.println(F("$PUBX,40,GLL,0,0,0,0*5C")); //GLL OFF
5140
5141    // Used as watchdog
5142    pinMode(13, OUTPUT);
5143
5144    //Cycle controle
5145    NoOfCycle = 0;
5146
5147    // initialize the timer chain
5148    TimerSchakel.TimerId = 0;
5149    TimerSchakel.Wait = 0;
5150    TimerSchakel.State = false;
5151    TimerKetting = new TimerType;
5152    *TimerKetting = TimerSchakel;
5153    MessageKetting = 0;
5154
5155    // do some initialization
5156    CConfigurationOK = true;
5157
5158    for (int i = StartDigitalPins; i <= EndDigitalPins; i++) LayoutDigital[i - StartDigitalPins] = false;
5159    for (int i = StartPWMPins; i <= EndPWMPins; i++) LayoutPWM[i - StartPWMPins] = false;
5160    for (int i = StartAnalogInPins; i <= EndAnalogInPins; i++) LayoutAnalogInPins[i - StartAnalogInPins] = false;
5161    //DigitalPins
5162    CConfigurationOK = ((FirstInput == 0) || (FirstInput >= StartDigitalPins));
5163    if (CConfigurationOK) CConfigurationOK = ((FirstInput + NoInputPins - 1) <= EndDigitalPins);
5164    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,FirstInput+NoInputPins-1)<=EndDigitalPins"));
5165    if (CConfigurationOK) CConfigurationOK = ((FirstOutput == 0) || (FirstOutput >= StartDigitalPins));
5166    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,FirstOutput>=StartDigitalPins"));
5167    if (CConfigurationOK) CConfigurationOK = ((FirstOutput + NoOutputPins - 1) <= EndDigitalPins);
5168    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,(FirstOutput+NoOutputPins-1)<=EndDigitalPins"));
5169    if (CConfigurationOK) CConfigurationOK = ((FirstUltra == 0) || (FirstUltra >= StartDigitalPins));
5170    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,FirstUltra>=StartDigitalPins"));
5171    if (CConfigurationOK) CConfigurationOK = ((FirstUltra + (2 * NoUltrasonic) - 1) <= EndDigitalPins);
5172    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,FirstUltra+(2*NoUltrasonic)-1)<=EndDigitalPins"));
5173    //AnalogInPins
5174    if (CConfigurationOK) CConfigurationOK = ((FirstAnaIn == 0) || (FirstAnaIn >= StartAnalogInPins));
5175    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,irstAnaIn>=StartAnalogInPins"));
5176    if (CConfigurationOK) CConfigurationOK = ((FirstAnaIn + NoAnaInPins - 1) <= EndAnalogInPins);
5177    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,(FirstAnaIn+NoAnaInPins-1)<=EndAnalogInPins"));
5178    //PWM Pins
5179    if (CConfigurationOK) CConfigurationOK = ((FirstAnaOut == 0) || (FirstAnaOut >= StartPWMPins));
5180    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,FirstAnaOut>=StartPWMPins"));
5181    if (CConfigurationOK) CConfigurationOK = ((FirstAnaOut + NoAnaOutPins - 1) <= EndPWMPins);
5182    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,(FirstAnaOut+NoAnaOutPins-1)<=EndPWMPins"));
5183    if (CConfigurationOK) CConfigurationOK = ((FirstServo == 0) || (FirstServo >= StartPWMPins));
5184    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,FirstServo>=StartPWMPins"));
5185    if (CConfigurationOK) CConfigurationOK = ((FirstServo + NoServos - 1) <= EndPWMPins);
5186    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,(FirstServo+NoServos-1)<=EndPWMPins"));
5187    if (CConfigurationOK) CConfigurationOK = (OneWireChannel <= EndPWMPins);
5188    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,(OneWireChannel <= EndPWMPins)"));
```

```
5189    //
5190    // if the configuration check is still ok, all start en ends are ok, now checking for dubbles
5191    //
5192    // Digital pins; Digital In, Digital Out and Ultrasonics
5193    if (CConfigurationOK) for (int i = FirstInput; i < (FirstInput + NoInputPins); i++) LayoutDigital[i - StartDigitalPins] = true;
5194    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for (int i=FirstInput; i<(FirstInput+NoInputPins); i++) LayoutDigital[i-StartDigitalPins] = true"));
5195    if (CConfigurationOK) for (int i = FirstOutput; i < (FirstOutput + NoOutputPins); i++)
5196    {
5197        CConfigurationOK = LayoutDigital[i - StartDigitalPins] == false;
5198        LayoutDigital[i - StartDigitalPins] = true;
5199        if (CConfigurationOK == false) break;
5200    };
5201    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for (int i=FirstOutput; i<(FirstOutput+NoOutputPins); i++)"));
5202    if (CConfigurationOK) for (int i = FirstUltra; i < (FirstUltra + (2 * NoUltrasonic)); i++)
5203    {
5204        CConfigurationOK = LayoutDigital[i - StartDigitalPins] == false;
5205        LayoutDigital[i - StartDigitalPins] = true;
5206        if (CConfigurationOK == false) break;
5207    };
5208    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for (int i=FirstUltra; i<(FirstUltra+(2*NoUltrasonic)); i++)"));
5209
5210    // Ananlog input pins; Only analogue inputs
5211    if (CConfigurationOK) for (int i = FirstAnaIn; i < (FirstAnaIn + NoAnaInPins); i++) LayoutAnalogInPins[i - StartAnalogInPins] = true;
5212    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for (int i=FirstAnaIn; i<(FirstAnaIn+NoAnaInPins); i++) LayoutAnalogInPins[i-StartAnalogInPins] = true"));
5213
5214    // PWM Pins; Analogue output, servo's and OneWire temperature sensors
5215    if (CConfigurationOK) for (int i = FirstAnaOut; i < (FirstAnaOut + NoAnaOutPins - 1); i++) LayoutPWM[i - StartPWMPins - 1] = true;
5216    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for (int i=FirstAnaOut; i<(FirstAnaOut+NoAnaOutPins); i++) LayoutPWM[i-StartPWMPins] = true"));
5217    if (CConfigurationOK) for (int i = FirstServo; i < (FirstServo + NoServos - 1); i++)
5218    {
5219        CConfigurationOK = LayoutPWM[i - StartPWMPins] == false;
5220        LayoutPWM[i - StartPWMPins] = true;
5221        if (CConfigurationOK == false) break;
5222    };
5223    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for (int i=FirstServo; i<(FirstServo+NoServos); i++"));
5224    if (CConfigurationOK)
5225    {
5226        CConfigurationOK = LayoutPWM[OneWireChannel - StartPWMPins] == false;
5227        LayoutPWM[OneWireChannel] = true;
5228    }
5229    if (!CConfigurationOK) Serial.println(F("%INF-CNFNOK-SETUP,for one wire channel"));
5230
5231
5232
5233    if ( CConfigurationOK == true )
5234    {
5235      // Start I2C bus
5236
5237      Wire.begin();
5238
5239      RTCAvailable = true;
5240      if (!RealTimeClock.begin())
5241         {
5242         Serial.println(F("%INF-RTC-UTS, Unable to sync with the RTC"));
5243         RTCAvailable = false;
5244         }
```

```
5245      else
5246         {
5247         Now = RealTimeClock.now();
5248         IntYear = int(Now.year());
5249         IntMonth = int(Now.month());
5250         IntDay = int(Now.day());
5251         IntHour = int(Now.hour());
5252         IntMinute = int(Now.minute());
5253         IntSecond = int(Now.second());
5254         setTime(IntHour, IntMinute, IntSecond, IntDay, IntMonth, IntYear);
5255         Serial.print(F("%INF-RTC-STM, RTC time is : "));
5256         digitalClockDisplay ();
5257         }
5258      Serial.println(F("%INF-CNFOK-SETUP, Configuration ok"));
5259      Serial.println();
5260      InitInputs();
5261      InitOutputs();
5262      InitAnaIn();
5263      InitAnaOut();
5264      InitServos();
5265      InitUltrasonics();
5266      InitTemp();
5267      InitMarkers();
5268      InitFSMStates();
5269      sensors.begin ();
5270      // Begin with state START
5271      TransitionToState("START");
5272      if (UseHMISerial) GMI();
5273
5274    }
5275
5276    //Interrupt handler, example
5277    //pinMode ( 20, INPUT );
5278    //digitalWrite ( 20, HIGH);
5279    //attachInterrupt ( Pin20, Isr, RISING );
5280
5281    //*============= START USER SETUP ======= START USER SETUP =====================
5282
5283    //Kp = 1;
5284    //Ki = 1;
5285    //Kd = 0.85;
5286    //Setpoint = 90;
5287    //Controller1.SetOutputLimits(0,180);
5288    //Controller1.SetTunings ( Kp, Ki, Kd );
5289    //Controller1.SetSampleTime ( 150 );
5290    //Controller1.SetMode(AUTOMATIC);
5291
5292    //*============  END USER SETUP  =======  END USER SETUP  ======================
5293
5294    Serial.print(F("HelpMonitor > "));
5295
5296  }
```