

```

/*
  L45AI2CPort.h
  als Include-Datei erstellt am 09.02.22

  In dieser Datei werden alle Definitionen und Funktionen abgelegt, die für die
  Verwendung
  des Horter I2C-Port über die I2C Schnittstelle erforderlich sind.
  Erste Test in Verbindung mit
  sketch_jan15b_controllinino_BasicShutters_Index

  */

byte I2CPortEingang[8];          // das am I2C-Port gelesene Eingangssignal

// 09.02.2022, neu entsprechend sketch_sep27b
void initI2C() {
  Wire.begin();                  // I2C-Pins definieren
  // setzten aller Bits der Eingabekarte auf 1
  // -----
  Wire.beginTransmission(I2C_IN_ADDR); // Start Übertragung zum PCF8574
  Wire.write(0xFF);                 // Alle Bits sind Eingänge
  Wire.endTransmission();           // Ende
} // Ende void initI2C()

void bit2byte(byte eingänge) {
  for (int i = 0 ; i < 8; i++) {
    if (eingänge & bit(i)) {
      I2CPortEingang[i] = HIGH;          // das am I2C-Port gelesenen
      Eingangssignal
    }
    else {
      I2CPortEingang[i] = LOW;
    }
  }
  /*
    Serial.print("I2CPort Bit am Eingang : ");
    Serial.print(i);
    Serial.print(" hat Wert : ");
    Serial.println(I2CPortEingang[i]);
  */
  // I2Centprell(i);
} // Ende for (int i = 0 ; i < 8; i++)
} // Ende void bit2byte(byte eingänge)

void parseBits(byte eingänge) {
  if (eingänge & bit(0)) {
    DEBUG_PRINTLN("I2C Eingang 1 für Bad auf 30% ");
  }
  if (eingänge & bit(1)) {
    DEBUG_PRINTLN("I2C Eingang 2 für Bad auf 60% ");
  }
  if (eingänge & bit(2)) {
    DEBUG_PRINTLN("I2C Eingang 3 für Schlafen auf 30% ");
  }
  if (eingänge & bit(3)) {
    DEBUG_PRINTLN("I2C Eingang 4 für Schlafen auf 50% ");
  }
  if (eingänge & bit(4)) {
    DEBUG_PRINTLN("I2C Eingang 5 für ArbeitWest auf 30% ");
  }
}

```

```

    }
    if (eingaenge & bit(5)) {
        DEBUG_PRINTLN("I2C Eingang 6 für ArbeitWest auf 80%");
    }
    if (eingaenge & bit(6)) {
        DEBUG_PRINTLN("I2C Eingang 7 für ArbeitSued auf 30% ");
    }
    if (eingaenge & bit(7)) {
        DEBUG_PRINTLN("I2C Eingang 8 für ArbeitSued auf 55% ");
    }

    DEBUG_PRINTLN(" ... ");
} // Ende void parseBits(byte eingange)


void leseI2CPort() {
    // Einlesen der Bits aus der I2C-INPUT Karte
    // -----
    static byte wert; // am 09.02.22 auf lokale Variable geändert
    static byte altwert;
    Wire.requestFrom(I2C_IN_ADDR, 1); // Ein Byte (= 8 Bits) vom PCF8574 lesen
    while (Wire.available() == 0) // Warten, bis Daten verfügbar
        ;
    wert = 255 - Wire.read(); // in invertierte Eingabe wandeln
    bit2byte(wert);

    if (wert != altwert) { // Wert nur ausgeben wenn er sich ändert
        DEBUG_PRINTLN_VALUE("neuer Wert: ", wert);
        // Wert auf "Serial Monitor" ausgeben
        DEBUG_PRINTLN_VALUE("   dezimal und binär: ", (wert, BIN));
        //Serial.println(wert, BIN);
        //printI2CPort();
        //test = !test;
        //digitalWrite(LED_BUILTIN, !test);
        parseBits(wert);
        Wire.endTransmission(true);
        // bit2byte(wert);
        altwert = wert; // und nachführen
    }
} // Ende void leseI2CPort()


void printI2CPort() {
    for (int i = 0 ; i < 8; i++) {
        DEBUG_PRINTLN_VALUE("I2CPort Bit am Eingang : ", i);
        DEBUG_PRINTLN_VALUE("   hat Wert : ", I2CPortEingang[i]);
    }
} // Ende void printI2CPort()

```