1.7.2024

Provenzano David, Yves Jaros

IMS-T

Things we don’t know about our code

Inhaltsverzeichniss

[ClockProvider.cpp 2](#_Toc170722613)

[CP\_init 2](#_Toc170722614)

[struct tm build\_time (7) 2](#_Toc170722615)

[struct 2](#_Toc170722616)

[tm 2](#_Toc170722617)

[build\_time 2](#_Toc170722618)

[strptime(\_\_DATE\_\_ " " \_\_TIME\_\_, "%b %d %Y %H:%M:%S", &build\_time); (20) 2](#_Toc170722619)

[strptime 2](#_Toc170722620)

[\_\_DATE\_\_" "\_\_TIME\_\_ 2](#_Toc170722621)

[%b 2](#_Toc170722622)

[%d 2](#_Toc170722623)

[%Y 2](#_Toc170722624)

[%H 2](#_Toc170722625)

[%M 2](#_Toc170722626)

[%S 2](#_Toc170722627)

[rtc.setTime(mktime(&build\_time) - offset); (21) 2](#_Toc170722628)

[setTime(…) 2](#_Toc170722629)

[mktime(&build\_time) 3](#_Toc170722630)

[LedWallEncoder.cpp 3](#_Toc170722631)

[uint8\_t (8) 3](#_Toc170722632)

[WallDriver.cpp 3](#_Toc170722633)

[WD\_BufferOutput 3](#_Toc170722634)

[\*buffer (19) 3](#_Toc170722635)

[digitalWrite 3](#_Toc170722636)

[(LA, LOW) (20) 3](#_Toc170722637)

[PixelPush 3](#_Toc170722638)

[digitalWrite(SDI, value > 0 ? HIGH : LOW) (38) 3](#_Toc170722639)

[Main.cpp 4](#_Toc170722640)

[const … (11) 4](#_Toc170722641)

[static … (17) 4](#_Toc170722642)

[Serial.begin(115200) (30) 4](#_Toc170722643)

[CP\_init(7200) (32) 4](#_Toc170722644)

[pinMode(…, …); (36 - 39) 4](#_Toc170722645)

# ClockProvider.cpp

## CP\_init

### struct tm build\_time (*7*)

#### struct

Struktur, die ein Kalenderdatum und eine Kalenderzeit enthält.

#### tm

Speichert Datum und Uhrzeit Informationen.

#### build\_time

Variable von struct tm.

### strptime(\_\_DATE\_\_ " " \_\_TIME\_\_, "%b %d %Y %H:%M:%S", &build\_time); (*20*)

#### strptime

Bildet einen String mit Datum und Uhrzeit den er dann in struct tm konvertiert

#### \_\_DATE\_\_" "\_\_TIME\_\_

Sind spezielle vordefinierte Makros, die den Zeitpunkt der Kompilierung des Programms enthalten.

#### %b

Monat

#### %d

Tag des Monats [1–31]

#### %Y

4-stelliges Jahr

#### %H

Stunde im 24-Stunden-Format [0–23]

#### %M

Minute [0-59]

#### %S

Sekunde [00-61]

### rtc.setTime(mktime(&build\_time) - offset); (*21*)

#### setTime(…)

Setzt die Uhrzeit der RTC (Echtzeituhr) auf die angepasste Zeit.

#### mktime(&build\_time)

Wandelt build\_time in Sekunden um ab dem 1 Januar 1970.

# LedWallEncoder.cpp

### uint8\_t (*8*)

«uint» seht für Unasigned Integer, 8 steht für 8 Bit und das «t» für «typedef».  
Die Variabel die dies bekommt kann nur positiv und nicht null sein.

# WallDriver.cpp

## WD\_BufferOutput

### \*buffer (*19*)

Buffer ist ein Pointer, der Stern weist darauf hin, dass es ein Pointer ist. Es ist der Index des Buffer Arrays.

### digitalWrite

«digitalWrite» ist die Funktion um die GPIO-pins (General Purpose Input Output) anzusteuern. dabei ist der erste Parameter die Adresse des Pins und der zweite der wert

#### (LA, LOW) (20)

LA ist einer dieser GPIO-pins der als einer einfach lesbarer variabel dargestellt wird.  
der Parameter Nr. 2 hat die Option «LOW» oder «HIGH» zu haben. dabei steht Low für einen Output von 0 V und High für 5 V oder 3,3 V Jeh nach Gerät.

## PixelPush

### digitalWrite(SDI, value > 0 ? HIGH : LOW) (*38*)

“value > 0 ? HIGH : LOW” ist eine einzeilige If-Abfrage.

# Main.cpp

### const … (*11*)

Ist ein Schlüsselwort bei der Deklaration von Variablen, die dafür sorgen, dass der Wert unverändert bar ist.

### static … (*17*)

Im Kontext zu Variablen sorgt «static» für «Lokale» Variablen, das bedeutet das die Variablen nur in dem lokalen File aufgerufen werden können

### Serial.begin(115200) (*30*)

Definiert die Bitrate für die Serielle Datenübertragung zu dem seriellen Monitor. Der Parameter ist die Anzahl Bits pro Sekunde.

## CP\_init(7200) (*32*)

“CP\_init” ist eine selbstgemachte Funktion, die mit dem Parameter ins Sekunden die aktuelle Zeit berechnet

### pinMode(…, …); (*36 - 39*)

Konfiguriert Pins entweder als Input oder als Output. Der erste Parameter ist der GPIO-Pin und der Zweite deklariert den Modus also entweder «INPUT» oder «OUTPUT».

#### OE, LA, …

Parameter 1, GPIO\_nr

#### OUTPUT, INPUT

Parameter 2