3a. Allgemeine Software - ESP32

Wir verwenden auf dem ESP32 die Software ESPEasy, um die Sensor-Rohdaten auszulesen und per MQTT an den LoxBerry weiterzuleiten. Natürlich könnt ihr auch Tasmota oder ESPHome o. ä. verwenden. Ihr müsst dann aber darauf achten, dass die Daten exakt so im MQTT Broker eingeliefert werden, wie es das Weatherstation Plugin später erwartet! Auch die Zykluszeiten (also die Sendeintervalle) müssen identisch sein!

Dann gibt es noch einen Unterschied, ob ihr den Blitzsensor AS3935 verwenden wollt oder nicht. Wenn ihr den Blitzsensor benutzen wollt, müsst ihr die Firmware selbst kompilieren (oder unsere vorkompilierte, aber eventuell etwas ältere Firmware verwenden). Ohne Blitzsensor könnt ihr die aktuellen vorkompilierten Versionen von ESPEasy verwenden, was das Ganze erheblich vereinfacht. Aber auch das Selbstkompilieren ist keine Raketenwissenschaft mit etwas PC-Erfahrung - ich gebe Euch hier eine detaillierte Anleitung.

Firmware ohne Blitzsensor AS3935

Folgende Software benötigt ihr:

• ESPEasy Firmware: https://github.com/letscontrolit/ESPEasy/releases

Zunächst ladet ihr Euch das neueste Release der Firmware herunter. Dieses findet ihr unter dem obigen Link unter "Assets". Wählt die Firmware aus, die für euren ESP32 passt - normaler die Standard-ESP32 Firmware, also z.B. ESPEasy_mega_20241222_ESP32_binaries.zip

Entpackt die Firmwarefiles mit einem ZIP-Programm in ein beliebiges Verzeichnis. Das war's an dieser Stelle schon. Macht jetzt im Kapitel Firmware flashen weiter.

Firmware mit Blitzsensor AS3935

Leider gibt es keine vorkompilierte Firmware mit ESPEasy, die alle von uns genutzten Sensoren beinhaltet. Deswegen muss die Firmware von uns selbst kompiliert werden. Keine Angst - so schwer ist das nicht. Wer es dennoch nicht hinbekommt, kann sich auch hier die von uns bereits kompilierte Firmware herunterladen. Diese ist vermutlich nicht mehr ganz neu, das macht aber in aller Regel gar nichts. So funktioniert einwandfrei. Download hier:

 $esp_easy_mega_20241231_custom_esp32_4m316k.zip$

Die Datei muss noch mit einem ZIP-Programm in ein beliebiges Verzeichnis entpackt werden. Macht dann im Kapitel Firmware flashen weiter.

Wenn ihr die Firmware selbst kompilieren wollt, benötigt ihr folgende Software:

- ESPEasy Source Code: https://github.com/letscontrolit/ESPEasy/archive/refs/heads/mega.zip
- Microsoft Visual Code Studio: https://code.visualstudio.com/

Als erstes ladet ihr Euch den Sourcecode vom ESPEasy Projekt über obigen Link herunter und entpackt diesen mit einem ZIP-Programm in ein beliebiges Verzeichnis.

Zunächst installiert ihr Microsoft Visual Code Studio auf eurem Rechner in startet es. Geht auf der linken Seite dann auf Extensions (der "kleine Würfel mit dem herausfliegenden Viertel" ganz unten). Nun sucht ihr der Reihe nach nach folgenden Extensions (über das Suchfeld) und installiert diese über den blauen Install Button:

- C/C++ Extension Pack (by Microsoft) installiert automatisch auch die Extension C/C++ mit
- **PlatformIO IDE** (by PlatformIO)



Die Installation dauert etwas - über ein kleines Fenster unten rechts könnt ihr jeweils den Status sehen. Nachdem die Installation durchgelaufen ist, müsst ihr VS Code einmal neu starten. Nun geht ihr in der rechten Leiste auf den Explorer und wählt den blauen Button Open Folder. Wechselt in das gerade eben entpackte Archiv mit dem Source Code. Ihr müsst das Verzeichnis auswählen, welches die Unterverzeichnisse src, lib, boards etc. enthält. Setzt den Haken bei Trust the author of all files in the parrent folder... und klickt auf Yes, ... Nun wird der Sourcecode von PlatformIO vorbereitet. Das dauert eine ganze Weile. Der Status wird Euch wieder unten rechts angezeigt.



Das war's an dieser Stelle. Macht jetzt im Kapitel Firmware flashen weiter.

Firmware flashen

Folgende Software benötigt ihr:

- Flash Download Tool: https://docs.espressif.com/projects/esp-test-tools/en/latest/esp32/production_stage/tools/flash_d ownload_tool.html
- PuTTY: https://www.putty.org/

Als erstes ladet ihr Euch noch die Flashsoftware "Flash Download Tool" herunter. Das Flash Download Tool wird auf euren Rechner mit einem ZIP-Programm entpackt. Jetzt verbindet ihr euren ESP32 per USB mit dem Rechner. Der ESP32 meldet sich als USB-to-UART Bridge CP210x oder FTDI am Rechner an. Sollte der Chip nicht erkannt werden, müsst ihr den entsprechenden Treiber noch installieren, siehe hier:

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/establish-serial-connection.htm l

Überprüft nun, welcher COM-Port eurem Device zugeordnet wurde: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/establish-serial-connection.htm l#check-port-on-windows

Wenn wir das alles vorbereitet haben, starten wir die Flashsoftware "Flash Download Tool". Wählt im Startscreen euren ESP32 aus (wenn er nicht gelistet ist, wählt ESP32) und setzt die Software auf Develop:

DOWN	-	×
ChipType:	ESP32]~
WorkMode:	Develop	~
LoadMode:	UART	~
	ОК	

Oben wählt ihr als erstes das Firmware-File aus, welches ihr flashen wollt. Nutzt ihr die fertig kompilierten Firmwares des ESPEasy Projekts, geht ihr dazu in das bin-Verzeichnis der entpackten ESPEasy-Firmware und sucht Euch die collection_G heraus. Diese beinhaltet alle Plugins die wir benötigen (aber eben ohne den Blitzsensor AS3935). Ihr müsst das **Factory**-File passend für euren ESP32 auswählen, also z. B.

ESP_Easy_mega_20241222_collection_G_ESP32_4M316k.factory.bin. Wenn ihr die selbst kompilierte Firmware nutzen wollt, dann findet ihr die gerade kompilierte Firmware im Verzeichnis build_output\bin. Auch hier müsst ihr das **Factory**-File auswählen.

Setzt vorne den Haken und gebt hinten als Startadresse hinter dem @-Zeichen 0x0 ein. Den SPI-Mode setzt ihr noch auf DOUT und wählt ganz unten den korrekten COM-Port aus, also z. B. COM3. Danach klickt ihr auf START und der Flashvorgang sollte starten. Er dauert ungefähr 2 Minuten.

×

Das war's. Macht jetzt im Kapitel Konfiguration weiter.

Konfiguration

Zieht den ESP32 vom USB ab und installiert und startet dann PuTTY. PuTTY stellt ihr auf eine Serielle Verbindung ein und gebt den COM-Port an, der eurem ESP32 von Windows zugewiesen wurde. Die Geschwindigkeit stellt ihr auf 115200.

Session	Basic options for your PuTTY session			
Logging	Specify the destination you wa	Specify the destination you want to connect to		
- Keyboard	Serial line	Speed		
Bell	COM3	115200		
- Features - Window - Appearance - Behaviour - Translation - Colours - Colours - Data - Prove	Connection type. <u>S</u> SH Serial (Other: Telnet		
	Saved Sessions	Load		
	and the second	Sa <u>v</u> e		
Serial Telnet Rlogin SUPDUP		Delete		
	Close window on exit	Only on clean exit		

Nun verbindet ihr den ESP32 wieder mit dem Rechner, wartet 1-2 Sekunden, bis der ESP erkannt wurde und klickt in PuTTY dann auf Open. Ihr solltet nun in PuTTY die Bootmeldungen des ESP32 sehen und er sollte das WLAN zur Erstkonfiguration aufmachen - die Meldungen sollten in etwa so aussehen:

Der ESP macht wie üblich ein eigenes WLAN auf - der Name variiert je nach Firmware etwas. Er lautet ESP_Easy, ESPEasy_Collection_G, lbweatherstation oder ähnlich. Das WLAN-Passwort ist configesp. Verbindet Euch mit dem WLAN und konfiguriert euer eigenes WLAN wie üblich. Anschließend startet der ESP neu und verbindet sich mit eurem eigenen WLAN. Wie immer findet ihr in eurem Router die IP-Adresse des neuen Gerätes heraus. Ruft die IP-Adresse in eurem Browser aus.

Unter Config vergebt ihr nun einen Unit-Namen (Hostnamen) und eine Unit-Nummer. Die Unit-Nummer ist beliebig, sollte aber eindeutig sein - sie dient der Kommunikation verschiedener ESPEasy Installationen untereinander. Wir nutzen dieses Feature allerdings nicht weiter. Vergebt ein Admin-Passwort für das Webinterface. Der Zugang ist dann durch den User admin mit passendem Kennwort geschützt. Den Rest lasst ihr hier auf den Standardeinstellungen.

ESP Easy Mega: Ibweatherstation_1					
₀Main oConfig OC	ntrollers 📌 Hardware 🕴 Devices 🔧 Tools				
Main Settings					
Unit Name:	Ibweatherstation				
	Note: Hostname: Ibweatherstation-1				
Unit Number:	1				
Append Unit Number to hostname:					
Admin Password:					
Wifi Settings					
SSID:	Ballyst.				
WPA Key:	••••				
Fallback SSID:					
Fallback WPA Kev:					

Im Tab Tools öffnet ihr nun noch unter System die Advanced Options. Unter Time Source aktiviert ihr NTP und nutzt den NTP-Server de.pool.ntp.org. Prüft die DST Settings (Sommer-/Winterzeit) und die Location Settings (hier die "Timezone Offset (UTC +)"). Unter Special and Experimental Settings aktiviert ihr noch "Restart WiFi Lost Conn", damit der ESP die WLAN-Verbindung automatisch neu aufbaut, wenn er sie verliert.

2025/04/16 01:57

oMain ⊖Config ⊖Contro	
	llers 📌 Hardware 🕴 Devices 🔨 Tools
Advanced Settings 1	
Rules Settings	
Rules:	
Tolerant last parameter:	
	Note: Perform less strict parsing on last argument of some commands (e.g. publish
SendToHTTP wait for ack	and sendToHttp)
SendToHTTP Follow Redirects:	
Time Source	
Use NTP:	
NTP Hostname:	de.pool.ntp.org
External Time Source:	None
DST Settings	
	Last v
Start (week, dow, month):	Sun v
	Mar
Start (localtime, e.g. 2h→3h):	2 [hour ~]
	Last v
End (week, dow, month):	Sun v
	Oct v
End (localtime, e.g. 3h→2h):	3 [hour •]
DST:	
Timezone Officet (LITC, +):	80 Iminuteal
Latitude:	0.00000 [2]
constitute.	0.000000 [1]
Longitude:	Note: Longitude and Latitude are used to commute summer and summer
Log Settings	
Syslog IP:	
Syslog UDP port:	514
Syslog Log Level:	None
Syslog Facility:	Kernel
Serial Lon Level	Info
Weblag land	hofe
Recial Consols Settings	into
Enable Serial Port Console:	
Baud Rate:	115200
Serial Port:	HW Serial0
ERD DY (DVO TY)	(000.3 N
ESP KA GPIU TA:	GPI0-3
ESP TX GPIO → RX:	GPI0-1
ESPEasy n2p UDP port	8286
Special and Experimental Setti	ngs
Webserver port:	80
	Note: Requires reboot to activate
Fixed IP Octet:	0
WD I2C Address:	0 (decimal)
I2C ClockStretchLimit:	0 [1/80 usec]
Enable Arduino OTA:	
Enable RTOS Multitasking:	•
JSON bool output without quotes:	
Collect Timing Statistics	
Enable RAM Tracker:	
Enable RAM Tracker: Allow TaskValueSet on all plugins:	
Enable RAM Tracker: Allow TaskValueSet on all plugins: Check I2C devices when enabled:	
Enable RAM Tracker: Allow Track/alueSet on all plugins: Check I2C devices when enabled: Allow OTA without size-check:	
Enable RAM Tracker: Allow TaskValueSet on all pligins: Check I2C devices when enabled: Allow OTA without size-check:	Note: When enabled, OTA updating can overwrite the filesystem and settings! Requires reboot to activate
Lando RAM Tracker: Enable RAM Tracker: Allow TaskValueSet on all plugins: Check I2C devices when enabled: Allow OTA without size-check: Web light/dark mode:	
Lando RAM Tracker: Enable RAM Tracker: Allow TaskValueSet on all plogins: Check I2C devices when enabled: Allow OTA without size-check: Web light/dark mode: Disable Rules auto-completion:	
Verbendten Enable RAM Tracker: Alkow TakValueSet on all plugini: Dickit 22 divides when anabled: Allow OTA without size-check: Web light/dark mode: Disable Rules auto-completion:	
Une num y unmethal Enable RAM Tracker: Alkow TakValueSet on all plugint: Dicki 22 devices when enabled: Allow OTA without size-check: Web light/dark mode: Disable Rules auto-completion: Disable Rules auto-completion: Disable Save Config es. tar:	
The Control of Control	
Les rend generation: Enable RAM Tracker: Altow Tark AlbueSet on all pignin: Orack I20 Grw/ces when enabled: Allow OTA without size-check: Web light/dark mode: Disable Rules auto-completion: Disable Save Config as tar: Use SSDP. Connection Pailure Threehold:	C C C C C C C C C C C C C C C C C C C
The Theorem of Constraints: Enable RAM Tracker: Alow Track ValueSet on all enabled: Chaoki 20 devices when enabled: New UTA without size-chock: Web light/dark mode: Disable Rules auto-completion: Disable Rules auto-completion: Disable Save Config as .tar: Use SSOP: Connection Failure Threshold: Force WiFi BJG:	
Learning demonstration: Enable RAM Tracker: Allow TAYAlueSet on all Julgina: Dock 22 devices when enabled: Disable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Disable Save Config as tar: Use SSDP: Connection Failure Threshold: Force WFI B/G:	
Enable RAM Took (Constraint) Enable RAM Took (Constraint) Allow Task/ValueSet on all plugin: Dook (22 divices when anabled: Meb light/dark mode: Disable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Connection Failure Threshold: Force WiFi No Sleep:	
Eachire RAM Tracker: Allow TrackValles of an all plugin: Dock (22 devices when enobled: Disable Rules auto-completion: Disable Rules auto-completion: Connection Failure Threshold: Force WiFi IbG: Parted WiFi No Sleep: Periodical send Grabulous APP:	
Line Time grammetrice: Emailer RAM Tracker: Adrow Track When Set on all pright: Check ECG Services when enabled: View lightidark mode: Disable Rules aufo-completion: Disable Save Config as tar: Uve SSDP: Connection Pailure Threshold: Force WiFi B/G: Remater WHFI Local Completion Force WiFi No Sitep: Periodical send Gratuitous ARPP: CPU Eco Mode:	C C C C C C C C C C C C C C C C C C C
Long Tenning Generalities: Enable RNA Tracker: Adrow Tarak ValvareSet on all program: Credit EQ Generalities when enabled: Instant ValvareSet on all credit EQ Generalities Allow OTA without size-check: Line SSDP: Connection Failure Threshold: Force WFI No Silesp: Periodical send Grabultous ARP- CPU Eco Mode:	Charge WIFI sleep settings requires reboot to activate Charge WIF
Learning generative: Enable RAM Tracker: Adrow Tark AlykueSet on all partial-RAM Tracker: Adrow Tark AlykueSet on all partial- Chard LGC devices when enabled: Three and Completion: Disable Rules auto-completion: Disable Save Config as tar: Use SSDP. Connection Failure Threehold: Force WIF1 BG: Rendart WIF1 DKC Exempt Periodical send Gratuitous ARP: CPU Eco Mode: Max WIF1 TX Power:	Chick Mane enabled. OTA updating can overwrite the filesystem and settings! Chick Also disables Rules syntax highlighting! Note: Also disables Rules syntax highlighting! Chick Change WiFi sleep settings requires reboot to activate Chick Change WiFi sleep settings Chick Chan
Leader RAM Tracker: Adox Tracker: Adox Tracker: Adox Tracker: Adox Track Adox Adox Adox Adox Adox Tracker: Ad	Context Mane anabled. OTA syndating can ownwrite the filesystem and settings! Proteives reacout to activate Prote: Also disables Rules syntax highlighting! Prote: Also disables Rules syntax highlighting! Prote: Change WIFF silesp settings requires reboot to activate Prote: Change WIFF silesp settings requires reboot to activate Prote: Change WIFF silesp settings requires reb
Learning demention: Enable RAM Tracker: Allow TAY AllueSet on all physics: Chaol 20 devices when enabled: Chaol 20 devices when enabled: Connection Failure Threshold: Force WIFI BAG: Restart WIFI DAS Learning Force WIFI BAG: Periodical send Grabilous ARP: CPU Eco Mode: Max WIFI TX Power: WIFI Stansitivity Margin:	
Learning demention: Enable RAM Tracker: Alow TarkalueSet on all plagan: Databack C. devices when enabled: Web lightidark mode: Disable Rules auto-completion: Disable Rules auto-completion: Disable Save Config as .tar: Use SSDP: Connection Failure Threshold: Force WiFi B/Q: Restart WiFi Lossi Conn; Force WiFi B/Q: Periodical send Gratulatou ARP: CPU Eco Mode: Max WiFi TX Power: WiFi Sanalityty Margin:	
Learning demendent Enable RAM Tracker: Altow Track values de on all pright: Check I2C Gw/ces when enabled: Web lightldark mode: Disable Rules aufo-completion: Disable Rules aufo-completion: Disable Save Config as Jar: Ure SSDP: Connection Pailure Threshold: Force WFI B/G: Remark WFI Garbailcos ARP: CPU Eco Mode: Max WFI TX Power: WFI Sensitivity Margin: Send WFI Max TX Power:	
Les renue generative: Enable RAM Tracker: Altow Tark Alvahue Bet on all pright: Check IZC GwYoss when enabled: Web lightidark mode: Disable Rules aufo-completion: Disable Rules aufo-completion: Disable Bave Config as Jan: Use SSDP: Connection Failure Threshold: Force WiFi BK2; Restant WiFi Loss Conn Force WiFi NS Siep: Periodical seend Grabulous ARP: CPU Eco Mode: Max WiFi TX Power: WiFi Sensitivity Margin: Send With Max TX Power: Extra WiFi scan loops:	Check When enabled. 0 TA updating can overwrite the filesystem and settingst Check When enabled. 0 TA updating can overwrite the filesystem and settingst Process and stables Rules syntax highlighting! Note: Abo disables Rules syntax highlighting! Note: Abo disables Rules syntax highlighting! Note: Change WHF silesp settings requires rebool to activate Note: Note may miss receiving packets with Eco mode enabled Note: Abo dimay miss receiving packets with Eco mode enabled Note: Change WHF silesp Settings requires rebool to activate Note: Change WHF silesp Settings requires rebool to activate Note: Abo dimay miss receiving packets with Eco mode enabled Note: About TX poaws to larged the AP with (sensibility + margin) dBm signal Note-repting University and the AP with (sensibility + margin) dBm signal Note repting University and the AP with (sensibility + margin) dBm signal Note Applied TX poaws to larged the AP with (sensibility + margin) dBm signal Note Applied TX poaws to larged the AP with (sensibility + margin) dBm signal
Learning demendent Enable RAM Tracker: Altow Tarki vlaueSet on all program: Check IZC devices when enabled: Allow OTA without size-check: Web light/dark mode: Disable Rules aufo-completion: Disable Save Config as .tar: Use SSDP: Connection Failure Threshold: Force WFI No: Siese: Periodical send Gratultous ARP: CPU Eco Mode: WFI Senaitively Margin: Send Wth Max TX Power: Extra WFI iscan loops:	Check Advant Ty bower to langed the AP with (sensitivity * margin) dBm signal Strengt Ty
Learning demendent Enable RAM Tracker: Allow Tarki value Set on all pagnatic: Chaok IZO devices when enabled: Allow OTA without size-chaok: Web light/dark mode: Disable Rules auto-completion: Disable Save Config as .sr: Use SSDP. Connection Failure Threshold: Force WIFI bG: Restart WEF ICAE Comm Periodical send Gratultous ARP: CPU Eco Mode: WIFI Sensitivity Margin: Max WIFI TX Power: Even WIFI Na TX Power: Even WIFI son toops: Use Last Connected AP from RTC:	Cate When enabled. O TA updating can overwrite the filesystem and settingal Cate When enabled. O TA updating can overwrite the filesystem and settingal Cate Also disables Rules syntaer highlighting(Cate Change WFF) sleep settings requires reboot to activate Cate
Learning demention: Enable RAM Tracker: Allow Tarkit AllueSet on all enabled: Chaoki 20 devices when enabled: Unable Save Config est Unable Rules auto-completion: Disable Rules auto-completion: Disable Rules auto-completion: Disable Save Config est as: Use SSOP: Connection Failure Threshold: Force WIFi BG: Restart WEF Loss Comm Force WIFi No Sleep: Periodical send Grabulous ARP: CPU Eco Mode: WIFi Sensitivity Margin: Sand WIPi Max TX Power: Extra WIFi TX Power: Extra WIFi Sciencingops: Use Last Connected AP from RTC: Datable Soft WIFi Auto Recomment:	Constrained on the second of
Land Terming Generation: Enable RAM Tracker: Adow Tra	Constraints of the second of
Les Trung Generation: Enable RAM Tracker: Altow Track Value Set on all Concell CG Gw/Ges when enabled: Und Start Value Set on all Charles CG W/Ge when enabled: Und Start Value Set on all Connection Failure Threshold: Force W/F IB/G: Remark W/FI Gold Conni Force W/FI No Silege: Periodical send Gratuitous ARP: CPU Eco Mode: W/FI Sensitivity Margin: Send W/FI No Silege: UNE Sensitivity Margin: Send W/FI No Silege: UNE Sensitivity Margin: Send W/FI No Silege: UNE Last Connected AP from Reconnect: Histion SSID Slow Connect:	Check When enabled, OTA updating can overwrite the filesystem and settingel Reverse reboot to activate Net: Also disables Rules syntar highlighting! Net: Also disables Rules syntar highlighting! Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change WHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate Net: Change MHF silesp settings requires reboot to activate to activate Net: Change MHF silesp settings requires reboot to activate to hi
Line Time generative: Enable RAM Tracker: Adrow Tark AlvahueSet on all partice. Adrow Tark AlvahueSet on all chark 120 Gawloss when enabled: Allow OTA without size-chack: Allow OTA without size-chack: Disable Rules aufo-completion: Disable Rules aufo-completion: Disable Save Config as Jar: Use SSDP: Connection Failure Threehold: Force WFI NB Siles: Periodical send Grabultous ARPP: CPU Eco Mode: WiFi Sensitivity Margin: Send WFI NX Power: WiFi Sensitivity Margin: Send WFI NA TX Power: WiFi Sensitivity Margin: List Adv Faultous ARPP: CPU Eco Mode: WiFi Sensitivity Margin: Send WFI NA TX Power: WiFi Sensitivity Margin: Histen SSID Silve Connect: Passive WFI Scan:	
Les Connected AP from Reading Automatical Enabler RAM Tracker: Altow Track Value Set on all product Convices when enabled: Altow OTA without size-check: Web light/dark mode: Disable Rules aufo-completion: Disable Save Config as Jar: Use SSDP: Connection Failure Threshold: Force WFI No Sleep: Periodical send Grabultous ARP: CPU Eco Mode: Max WFI TX Power: Extra WFI scantcoopt: Use Last Connected AP from MFC. Send Wth Max TX Power: Extra WFI scant: Use SSDP: Enable SXV WFI Auto Periodical Silo Slow Connect: Pasave WFI Scant: Pasave WFI Scant:	Construction of the source of the part with the file system and setting at the part of the source of the part of the

Im Tab Controllers fügen wir einen neuen Controller hinzu (bzw. editieren den bereits unter 1 angelegten Controller). Controller dienen der Kommunikation - wir richten hier unseren MQTT Broker ein. Als Protokoll wählt ihr Home Assistent (openHAB) MQTT aus und gebt die IP-Adresse und Port eures MQTT Brokers ein (normalerweise die IP-Adresse euer LoxBerry und Port 1883). Weiter unten unter Credentials aktiviert ihr "Use Extended Credentials" und gebt User und Passwort für euren MQTT Broker ein. Die notwendigen Einstellungen findet ihr im LoxBerry im MQTT Widget. Unter "MQTT" setzt ihr nun noch folgende Einstellungen:

- Controller Subscribe: %sysname%/sensors/#
- Controller Publish: %sysname%/sensors/%tskname%/%valname%
- Controller LWT Topic: %sysname%/status
- LWT Connect Message: running
- LWT Disconnect Message: stopped
- Send LWT to Broker: Ja
- Enabled: Ja

×

Anschließend sollte sich der ESP mit eurem MQTT Broker verbinden. Prüft den Status im Config Tab (gebt ihm ein paar Sekunden um sich zu verbinden).

Unter Hardware konfigurieren wir nun noch das I2C Interface und unsere GPIOs, die wir verwenden wollen. Zunächst konfigurieren wir unter I2C Interface die beiden GPIOs für SDA und DCL:

- GPIO SDA: GPIO-21
- GPIO SCL: GPIO-22

Danach setzen wir noch die GPIOs, die wir als digitale Eingänge verwenden wollen, als Eingang mit aktiviertem PullUp-Widerstand:

- GPIO-25: Input pullup
- GPIO-26: Input pullup
- GPIO-27: Input pullup
- GPIO-33: Input pullup

9/10

ESP Easy Mega: Ibweatherstation_1

oMain ⊚Config ⊜Controllers 📌 Hardware 🕴 Devices 🔧 Tools

Wifi Status LED		
GPIO → LED:	- None -	¥
Inversed LED:		
	Note: Use 'GPIO-2 (D4)' with 'Invers	ed' checked for onboard LED
Reset Pin		
GPIO - Switch:	- None -	~
I2C Interface	Note: Press about 10s for factory re-	sel
GPIO ≓ SDA:	GPI0-21	~
$GPIO \rightarrow SCL:$	GPIO-22	v
Clock Speed:	400000 [Hz]	
	Note: Use 100 kHz for old I2C devic	es, 400 kHz is max for most.
Slow device Clock Speed:	100000 [Hz]	
SPI Interface		
Init SPI:	Disabled	~
	Note: Changing SPI settings require on!	s to press the hardware-reset button or power of
GPIO host states	Note: Chip Select (CS) config must	be done in the plugin
Pin mode GPIO-0 A:	Default	~
Pin mode GPIO-1-	Default	V ITX0
Dis mode CDIO 2 Av	Defection	. [170]
Pin mode GPIO-2 A:	Default	
Pin mode GPIO-3:	Default	~ [RX0]
Pin mode GPIO-4:	Default	*
Pin mode GPIO-5:	Default	u u
Pin mode GPIO-12 A:	Default	×
Pin mode GPIO-13:	Default	~
Pin mode GPIO-14:	Default	¥
Pin mode GPIO-15 쇼:	Default	*
Pin mode GPIO-16:	Default	~
Pin mode GPIO-17:	Default	*
Pin mode GPIO-18:	Default	v
Pin mode GPIO-19:	Default	~
Pin mode GPIO-21:	Default	~ [I2C SDA]
Pin mode GPIO-22:	Default	- [12C SCL]
Pin mode GPIO-23:	Default	
Pin mode GPIO-25:	Input pullup	
Pin mode GPI0-26:	Input pullup	
Pin mode CPIO 97	Input pullup	
	nput putup	
Pin mode GPIO-32:	Default	
Pin mode GPIO-33:	Input pullup	
Pin mode GPIO-34 ⇔:	Default	•
Pin mode GPIO-35 ⊂:	Default	•
Pin mode GPIO-36 ←:	Default	v
Pin mode GPIO-37 ⇔:	Default	~
Pin mode GPIO-38 ⊂:	Default	~
Pin mode GPIO-39 ⇐:	Default	~

Powered by Let's Control It community

Build: ESP_Easy_mega_20241220_custom_ESP32_4M316k Dec 20 2024

Damit ist die Grundkonfiguration abgeschlossen! Die einzelnen Eingänge und Sensoren folgenden dann in den jeweiligen Unterkapiteln.

From: https://wiki.loxberry.de/ - LoxBerry Wiki - BEYOND THE LIMITS

Permanent link:

https://wiki.loxberry.de/howtos_knowledge_base/loxberry_wetterstation/3_software/esp32?rev=1735649814

Last update: 2024/12/31 13:56