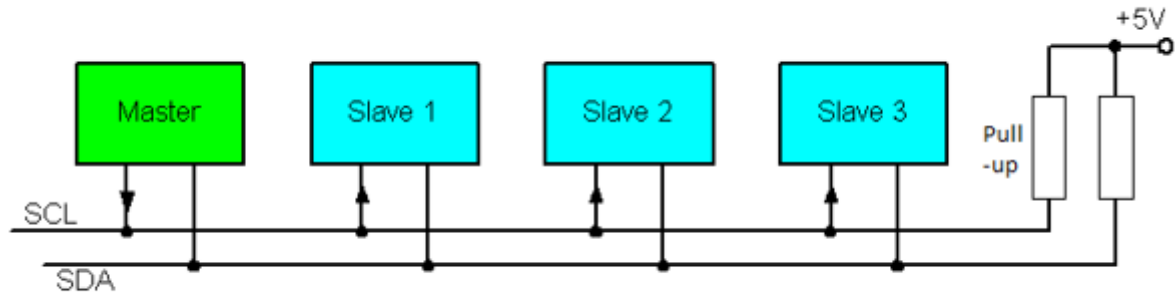


I2C tengibraut (bus):

Fyrir hluti sem þurfa ekki mikinn hraða getur verið ákjósanlegt að tengja saman á bus. Þetta geta verið hlutir eins og rofar og skjáir. T.d. er algengt að 16x2 lcd skjáir séu útbúnir fyrir tengingu á I2C tengibraut, en það fækkar línur frá skjá til örgjörva niður í tvær. Fleiri nærliggjandi skynjarar og hlutir geta svo tengst inn á tengibrautina.

Uppsetning I²C:

Mynd 1 fengin af <http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1163405566/2>

Yfirlitt er internal pullup nóg þegar t.d. bara einn hlutur er tengdur við örgjörva.

Debug Mode:

Til er aðferð til að stýra þýðingu á kóða yfir í vélamál. Hægt er setja #if #endif setningar utan um blokkir til þess að virkja eða óvirkja eftir stillingu.

Ein hagnýting á þessu er að hafa debug hluta sem hægt er að kveikja og slökkva á. Bæði kosta skilaboðin í hraða og minni (stýringin er pre-compile). Svo er verra þegar notendum er drekkt í skilaboðum. (sjá t.d. hér á Arduino forum: <https://forum.arduino.cc/index.php?topic=46900.0>)

Tengið EPS32 við I2C skjá

1. Tengið ESP32 við I2C skjá.
2. Bræðið saman stýringu fyrir skjáinn og hafið debug I2C scanner hluta fremst sem hægt er að kveikja og slökkva á með #define

ESP32 í sleep mode:

Hægt er að setja ESP32 í sleep mode. Til er bæði léttur svefn og djúpur svefn, en þá er slökkt á mismikið af innviðum stýriörgjörvans. Flettið upp straumnotkun, mælið hana og áætlið hversu lengi ESP32 endist á einni 18650 lithium cellu.

- a) Setjið ESP32 í sleep mode sem vaknar á tíu sekúndu fresti, tekur mælingu af pinna og sendir yfir Serial til tölvunnar og fer svo að sofa aftur.
 - Active Duration: Tími sem það tekir að lesa og senda yfir serial sem er um 1 sek
 - Active current: straumnotkun er um 20mA þegar tækið er í notkun.
 - Sleep duration: tíminn sem það tekur tækið að fara í sleep state og að vakan aftur sem er um 10 sek.
 - Sleep current: straumnotkun er um 7μA
 - Rafhlöðugeta: 2500mAh
 - notkun = $(20m + 7μ) / 11sek = 1.82m/cycle$
 - ending $2500m / 1.82m = 1373.5klst$
- b) Setjið ESP32 í sleep mode sem vaknar við að ýtt sé á takka og sendir „vaknadur“ á Serial til tölvunnar.
 - ESP32 sleep currnet: Straum notkun tækisins í svefni fer eftir ýmsum hlutum t.d uppsetningu aflgjafa stoðu pinna og hvaða svefn það er notað. Nákvæmasta svarið fæst í data sheet fyrir kubbin og önnur skjöl
 - Rafhlöðugeta: Þú þarft að vita afkastagetu rafhlöðunnar sem þú ert að nota, venjulega mæld í milliamper-stundum (mAh).
 - Meðalstraumnotkun: Þetta gildi táknar meðalstrauminn sem ESP32 dregur í virkri stillingu (þegar ekki í svefnstillingu). Þú getur mælt þetta gildi með því að nota margmæli eða vísað í ESP32 gagnablaðið eða einingarskjölin.
 - Vinnulota: Vinnulotan er hlutfall þess tíma sem ESP32 er vakandi og eyðir orku á móti tímanum sem hann eyðir í svefnstillingu. Þetta fer eftir því hversu oft þú ýtir á takka og vekur ESP32.
 - Meðalstraum notkun á 50% duty cycle: $(20m + 7μ) / 2 = 10mA$
 - Rafhlöðuending: $Rafhlöðugeta / (meðalstraumnotkun * vinnutími) = 2500m / (10m * 0.5) = 499.90klst$
 - áætlaður rafhlöðuending á 2500mAh cellu og 50% duty cycle er um 499.9klst
- c) Hripið niður smá pælingar í hverju munurinn liggur í mæligildi fyrir brettið og uppgefið fyrir örgjörva.

Munur á mælingum og áætlun í data sheet er ekki eins vegna þess að framleiðslan hefur sérstaka ramma sem kubburinn þar að vera inni þannig ekki allir kubbar eru eins of kallað silicom lottery

STTÆ3RE05BR

Verkefni 2b

Höf: Hákon Hákonarson

Hér er um sleep mode hjá espressif

(https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/sleep_modes.html)

Deep sleep má finna á ýmsum stöðum td. hér á Instructibles:

(<https://www.instructables.com/id/ESP32-Deep-Sleep-Tutorial/>)

Random nerds:

<https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/>

