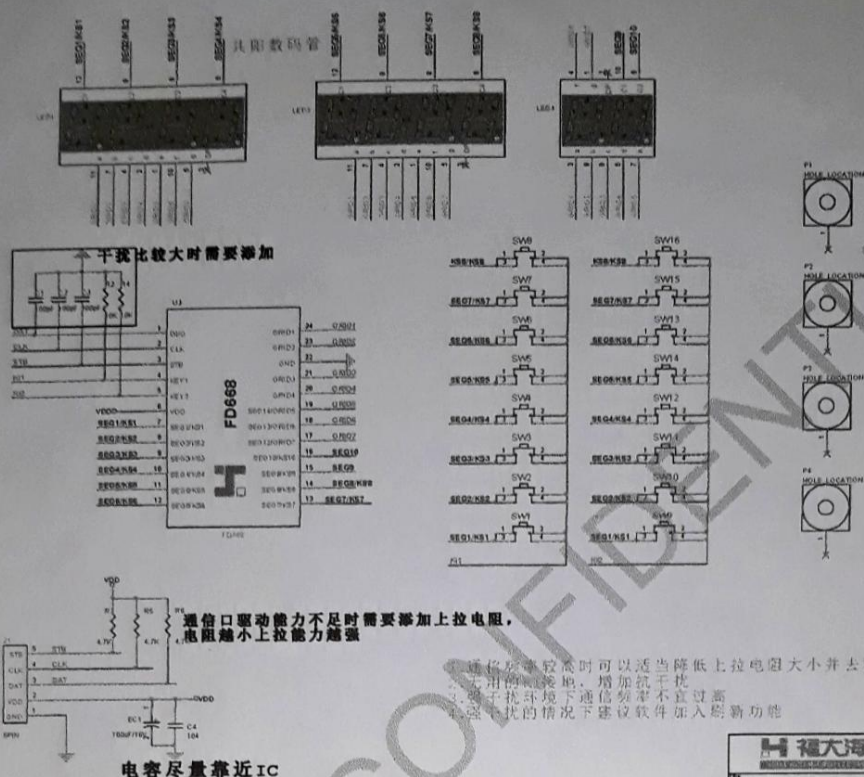


### 3. 驱动共阳数码管



### 4. 使用注意事项

1. 无用的 I/O 接地，增加抗干扰
2. 推荐客户推挽输出，高阻输入，这样高速通信的波形会更好
3. 高速通信的时候去除通信口上的电容，减小通信口的上拉电阻
4. 建议通信速度 100K 以下
5. 强干扰环境下通信频率不宜过高,建议软件加入刷新功能

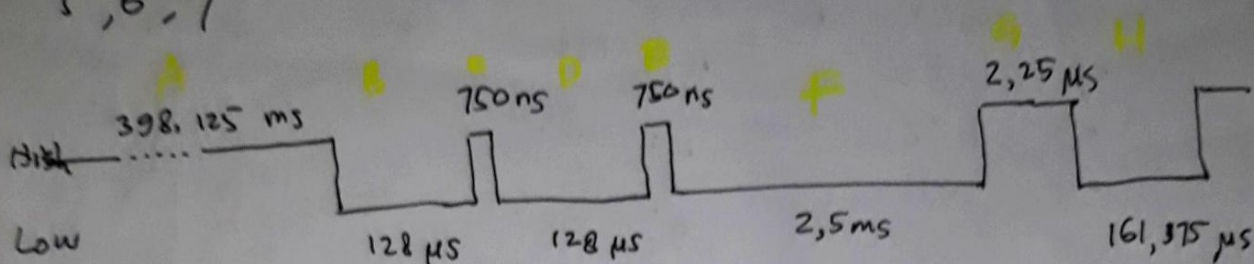
Weight display

FD668

BEAFBCAD01

2 } power  
3 }

5, 6, 7



$$128.000 \text{ ns} \quad 128 \mu\text{s} = 0.128 \text{ ms} = 0.000128 \text{ s}$$

nano  
 $10^{-9}$

micro  
 $10^{-6}$

milli  
 $10^{-3}$

$$750 \text{ ns} \quad 0.750 \mu\text{s} = 0.000750 \text{ ms} = 0.000000750 \text{ s}$$

$$0.0025 \text{ ms} = 0.0000025 \text{ s}$$

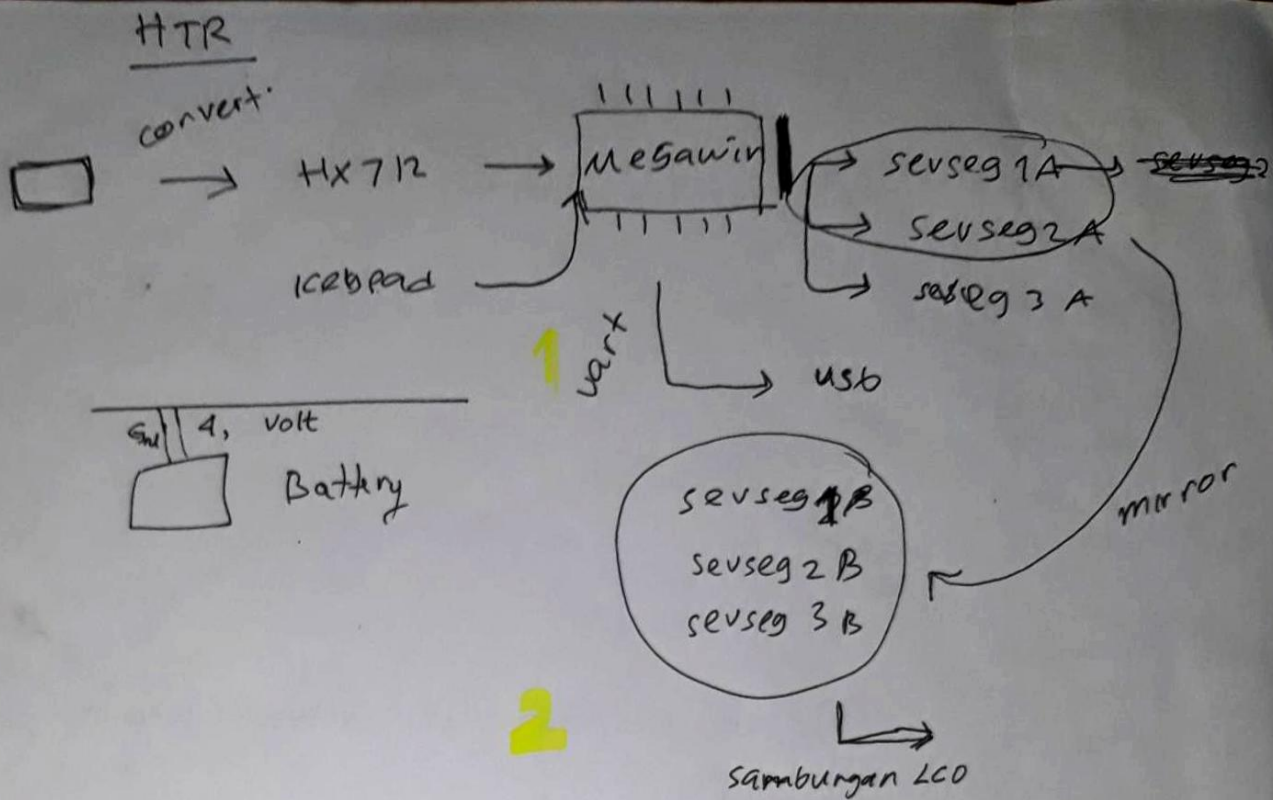
$$2.25 \text{ Ms} = 0.00225 \text{ s}$$

$$0.000161375 \text{ s}$$

$$\begin{array}{r} 80 \mu\text{s} \\ 80,000 \text{ ns} \\ ,800 \end{array}$$

$$\begin{array}{r} 875 \\ 750 \\ \hline 125 \end{array}$$





→ HX711 → Arduino nano → sevseg

→ HX712 → Megawin → esp

IC keypad

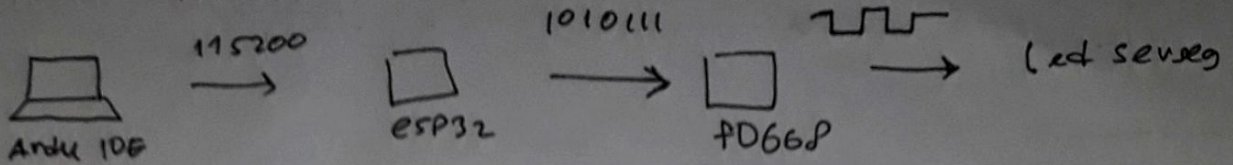
UART-RX99 megawin di mana?

Megawin	
RST	: 0, 9 V
COA	: 2, 5 V
VCC	: 3, 3 V
SCA	: 3 V
PB	: 0 V
RX	: 3, 3 V
TX	: 3, 3 V

1  
2  
3  
STB 4  
DIO 5  
CLK 6  
7  
8

Kabel Mirror	
<del>VCC</del>	: 3, 3 V Oket Bata
GNP	:
VCC	: 3, 3 V
	: 3 V
	: 2, 5 V — Pin bulat
	: 3 V
	: 3 V
	: 0 V

write process



shiftOutMod di program di Arduino IDE, supaya ESP32 bisa menghasilkan binary yg melalui mekanisme di dalam F0668 diubah menjadi Bit-Pulse dan ditampilkan di sensor.

delay	Bit/s	duration
100	10	950 ms
50	20	475 ms
25	40	237,5 ms
12,5	80	118,75 ms
6,5	158-160	60,126 ms





0106 h  
 0064. 6.  
 0115 5  
 0119 :5.  
 0123 :6

0110 h.  
 0121 E  
 298 n (kecil)  
 361 F.  
 365 F.

115200

57600

32400

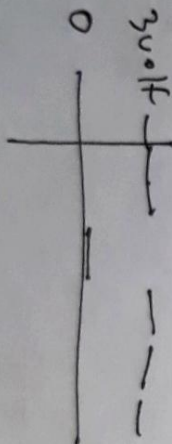
19200

9600

4800

2400

1200



0130 7  
 0134 7.  
 0182 3.

185 2  
 189 2.

152 8.

219 0

223 0.

227 9

230 9.

242 9

246 9.

372 :5.

376 :6.

378 :6

387 :7

388 :7.

130 .1

00:56 4.

01:36 1



Fix

243 9

247 9.

123 6

185 2

183 3.

130 1

49 0

195 7

199 7.

115 5

119 5.

223 0.

1341 9.

179 3

127 6.

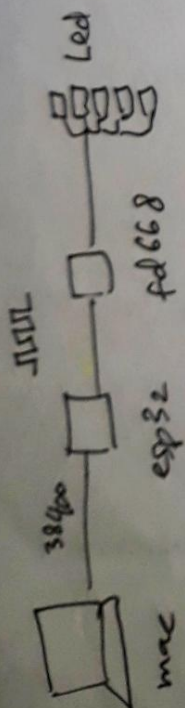
226 4

230 4.

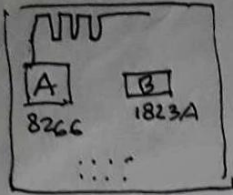
255 8.

251 8.

0  
= 1101

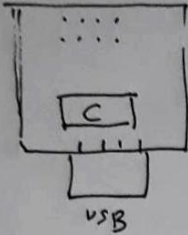


# HTR Lolin

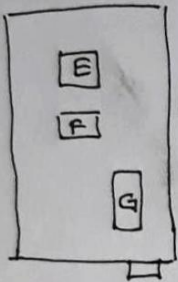


[A] wifi module

[B] serial (rx, tx)



[C] serial to usb



Lolin

[E] chip esp32 (wifi module + memory + 4G)

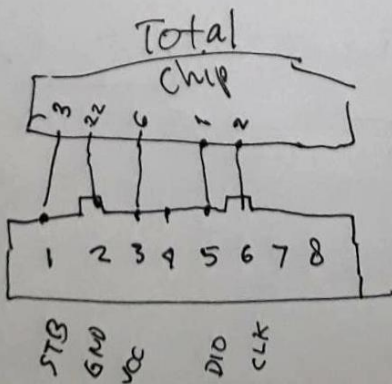
[F] serial

(CPU 2 core + ...)

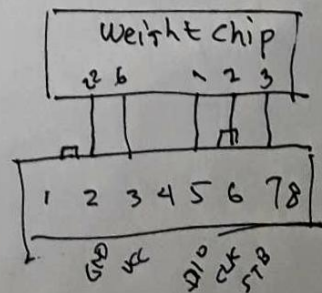
[G] serial to usb



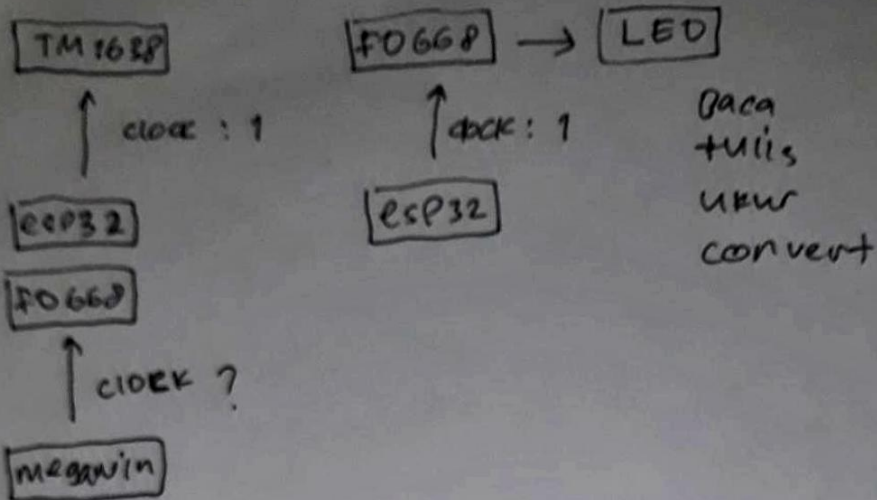
TX	EN	RST	3V3
GND	IO2	IO0	RX



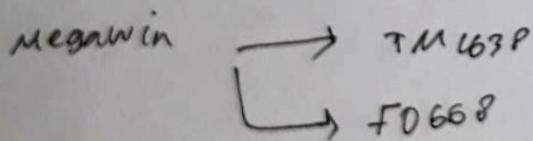
cable



cable



1. TM1638 & F0668 mirip frekuensi bitratanya. Bedanya TM1638 bisa read-write, f0668 cuma bisa write.
2. Megawin dikit transaksi. Dingin chipnya. Bitrate kemungkinannya kecil (nggak sampe jutaan).



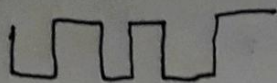
3. Megawin kalau dibaca pake digitalread esp, maka akan didapatkan tabel yg dikehendaki  
 → cari bitrate pakai sakae

049a



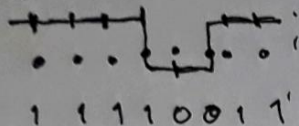
ubah digits

STB

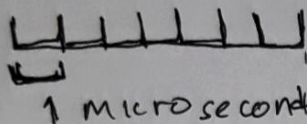


STB

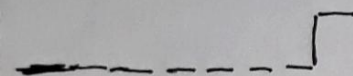
#2 1111 0011  
DI/O



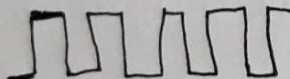
angka yg akan ditampilkan



#1  
STB



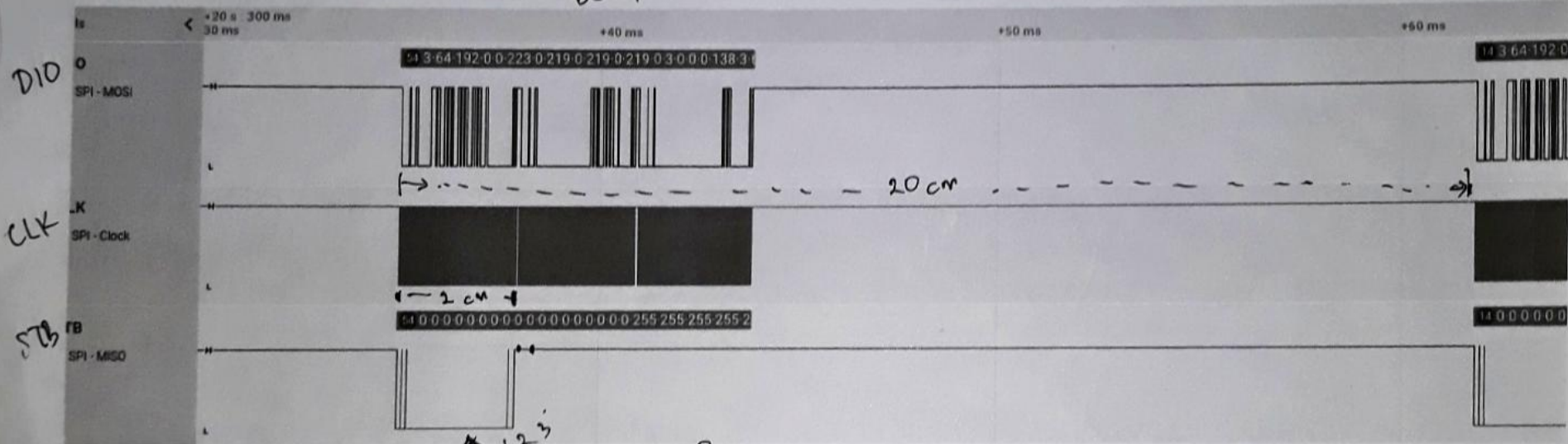
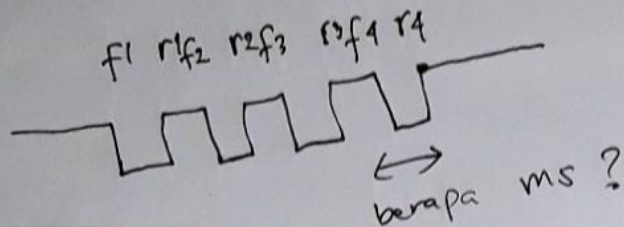
#3  
CLK



berapa cepat data <sup>di kirim</sup> ~~ditampilkan~~



$$\frac{6}{20} = \frac{3}{10} = 0,3$$



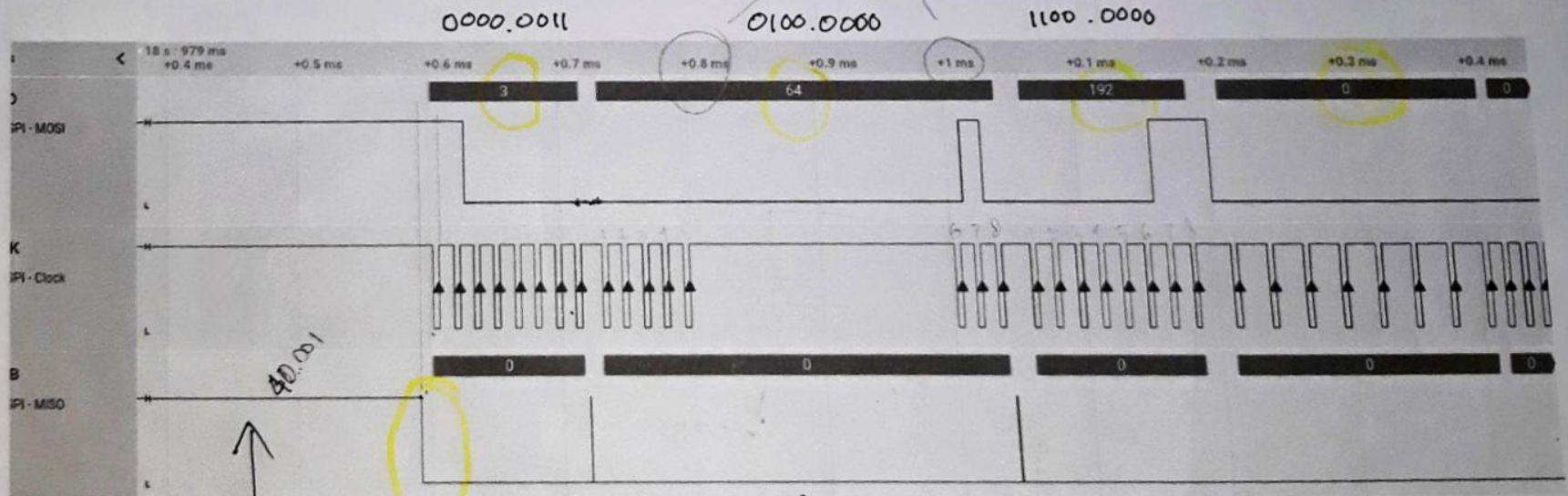
Handwritten notes and annotations:

- 123
- on Rising ( )
- d = digital Read (STB);
- HIGH, HIGH
- delay MS (5);
- rising = 0

Interrupt  
digital read

Bayangkan gambar ini sebagai video  
yg bergerak sesuai waktu.

3, 64, 192, 0, 0, 223



f1

r1 f2

r2 f3

1 2 3 4 5 6 7 8  
223 = 11111011

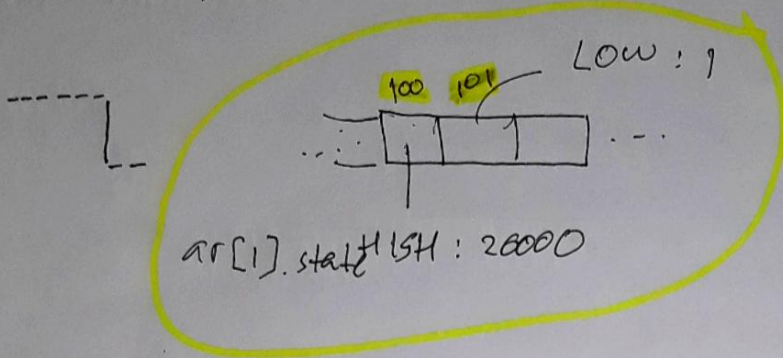
4 x 0.1

Test



Where's he going with this?

STB HIGH > 20000 dan falling pertama ✓



PIN → state HIGH : 20000

LOW : 1.

CLK diambil ketika HIGH rising

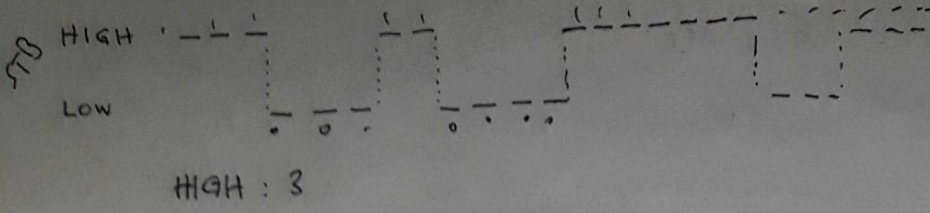
CLK rising, ambil 010 saat itu.

CLK punya fungsinya sendiri, STB punya fungsinya sendiri.

/\* ...

\*/  
void func() {

STB + true, mulai → ambil data  
CLK



array of uint8\_t

Pembandingan antara 2 data untuk memonitora posisi

- 1 HIGH : 3
- 2 FALL : 1
- 3 LOW : 3
- 4 RISE : 1
- 5 HIGH : 2
- 6 FALL : 1
- 7 LOW : 4
- 8 RISE : 1
- 9 HIGH : 7
- 10 FALL : 1
- 11 LOW : 3
- 12 RISE : 1
- ~~RISE : 1~~
- 13 HIGH : 3

loop dari awal sampai akhir

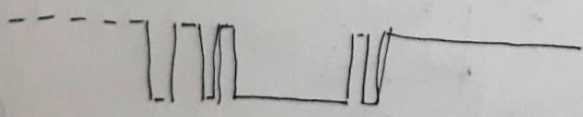
$$1 \mu\text{micro} = 1 \text{ s}$$

$$10 \mu\text{micro} = 10 \text{ s}$$

$$10.000.000 / 1 \text{ s}$$

$$10 / 1 \mu\text{s}$$

$$40 / 4 \mu\text{s}$$



$$\frac{10.000.000 \text{ read}}{1 \text{ sec}} : \frac{10.000.000}{1.000.000} = \frac{10 \text{ read}}{1 \text{ micro sec.}}$$

$$\frac{10}{10} = \frac{10 \text{ read}}{1000 \text{ ns}} = \frac{10}{10} = \boxed{\frac{1 \text{ read}}{100 \text{ ns} / 0.1 \mu\text{s}}}$$

$$\frac{1 \text{ read}}{0.1 \mu\text{s}}$$



4 us

1 data 0.1 us  
40 data 4 us

$10^3 = \text{kilo} = \text{ribu}$

100 ns = 0.1 us

50 ns =

$$0,1 = \frac{1}{10} = 10^{-1}$$

$$0,001 = \frac{1}{1.000} = 10^{-3} = \text{per ribu} = \text{mili}$$

$$0,000.000,1 = \frac{1}{1.000.000} = 10^{-6} = \text{per juta} = \text{mikro}$$

Skala

Perbandingan perasaan.

$10^6$

Madu

$10^4$

Box

$5 \cdot 10^8$

500.000.000

1.000.000

$500 \cdot 10^6$

+

$\uparrow$   
x  
2  
10  
yit.

Panjang Jarak

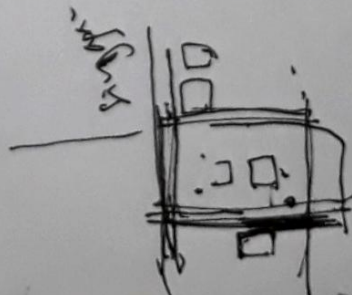
Sifat Inggil

Stafitno  
Woto  
Sumardi  
Ndot  
Ti-

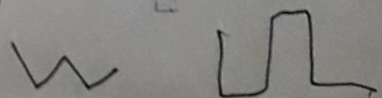
Amang  
25  
18  
0.1

010001110

100111



Output



1. data valid ketika: ~~stb berubah ke low~~  
gabungan antara clk HIGH dengan  
apitan STB Low \*EP: End-point \*SP: starting  
point  
Versi #1  
peristiwa pertama adalah STB low, tandai sebagai \*SP  
kemudian CLK naik turun. Ketika CLK naik, hitung  
sebagai 1 bit valid, ketika STB mulai high,  
catat sebagai \*EP. Saat itu harusnya var yg  
menampung data dari clk sudah menampung  
bit secara utuh untuk menggambar angka.

Versi #2 Bhs Ind

- ① STB Low yg pertama catat sebagai starting point  
dari proses sniff
- ② CLK akan naik turun di antara STB dan D/I/O.  
Ketika CLK HIGH, ambil datanya ~~dan simpan~~ dari  
D/I/O dan simpan.
- ② Tunggu STB sampai sudah berjalan selama  
5 kali.

C:

Koreksi langkah 2 : tidak perlu tunggu sampai  
5 kali, langsung operasikan langkah ③

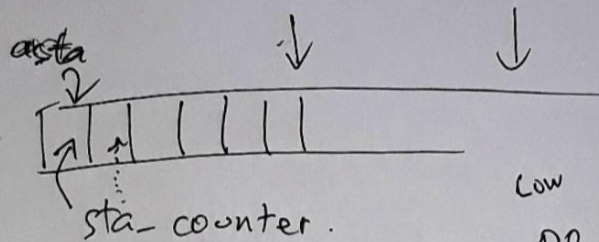


- Ambil data dari 01/0 ketika CLK ~~high~~ rising dan STB sudah masuk HIGH yg ke 200.000.

① Baca STB sampai ketemu high 200.000

- harus tahu ~~apa~~ data sebelumnya.
- hitungnya gimana? sehingga falling pertama ~~tan~~ diketahui sebagai starting point

④. Mirip kemarin, tapi nggak dimasakin ke array



Low R1 FAL HIGH BLANK  
00, 01, 10, 11, 100

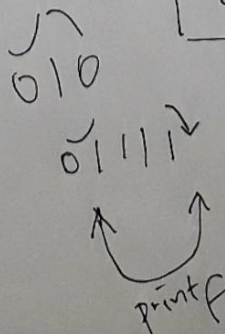
① geser sta-counter ke elemen array asta  
kalau input baru beda dg sebelumnya.

count, shift + state,

1 | 0 | 0 | 1 | 1

harus ada yg  
diabaikan supaya  
ringan

proses belajar kadang  
cepat & lambat



hippie (+) free thinker

3h/m adalah alat  
untuk zoom - zoom +

Apakah algoritma untuk menentukan apakah STB sedang diproses di tengah atau belum?

Tujuan:

- langsung baca data dari saluran STB/CLK/DIO
- problem utama adalah menentukan dimana posisi pembacaan data.

① hitung jumlah falling STB

① hitung jumlah HIGH STB, minimal 5, sebelum falling pertama.

② mulai hitung falling STB

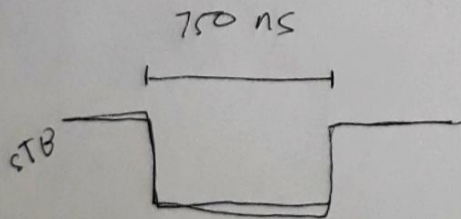
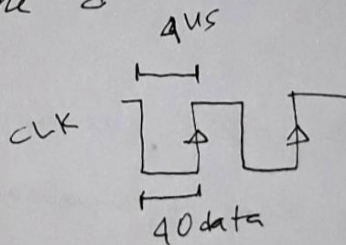
Where Where's he o  
What.

$$1 \mu s = 1000 \text{ ns}$$

$$40 \text{ data} / 4 \mu s$$

$$\frac{4.000 \text{ ns}}{100 \text{ ns}} = 40 \text{ read}$$

$$\frac{4 \mu s}{0,1 \mu s} = 40 \text{ read.}$$



$\frac{1 \text{ read}}{100 \text{ ns}} \rightarrow$  Untuk melakukan 1 read butuh waktu  
~~100 ns~~ 100 nano second. (0,1 μs)

$$750 / 100 = 7,5$$

$$= 7 \text{ data}$$

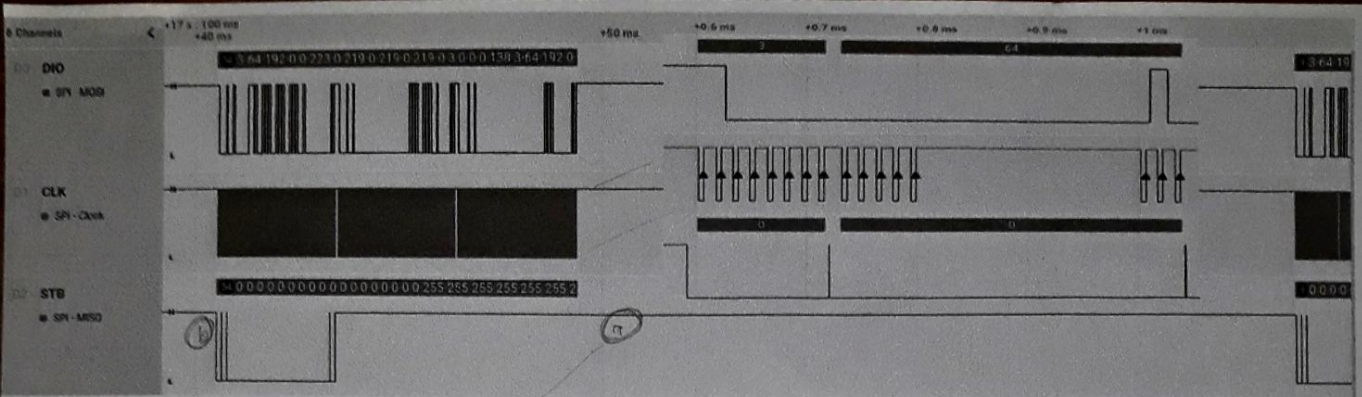
240

$$250 \text{ decimal} = 1111 1010$$

$$\begin{array}{r} 1111 1010 \\ + 10 \\ \hline 1111 1111 \\ + 1 \\ \hline 1111 1111 \end{array}$$

255





1 cycle data = 20 cm

(5)

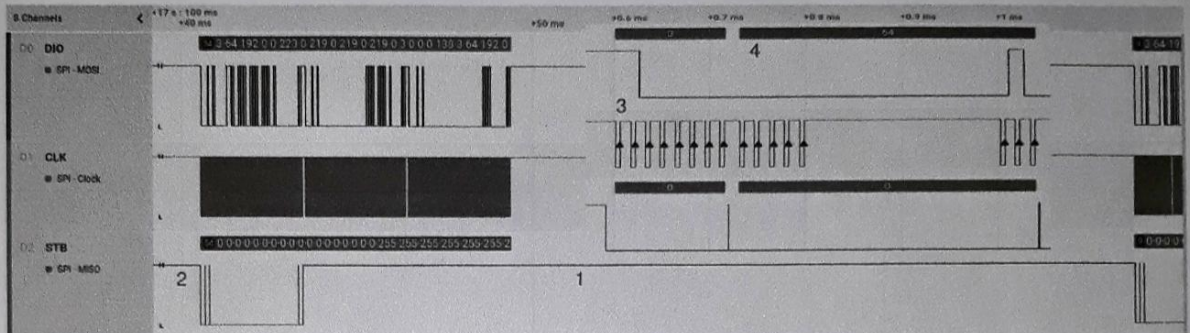
- 1 STB, CLK, dan DIO sudah, di tahap ini, bertepatan dengan baik. Data dari DIO sudah diambil, dan dimunculkan. Beberapa data ditampilkan secara benar, sisanya masih kurang tepat.
  - ~~di DIO~~
- 2 Selama ini sudah memproses Fapan data dibaca, apanya masih kurang.
  - a. cari DIO yg menghasilkan 3/64/192, DIO valid dihitung sejak itu.
  - b. Diantara 192 dan 138.
3. DIO diambil sampai STB ~~akhir~~ Rising terakhir.

## Dari Source-code, ke Paragraf, sampai Gambar

Input: 3 line dari megawin (STB, CLK, DIO)

Output: data desimal yang muncul di sevseg

Proses adalah sebagai berikut



Baca STB HIGH, catat jumlahnya. Ketika jumlah STB HIGH yang lalu itu >50000 (#1), dan sedang FALLING (#2), tandai sebagai ~~sempai~~ awal mula CLK diproses.

```
if (stb_pin_prev.ctr > 50000 && stb_pin.state == FALLING){
    shbf_done = 1;
```

```
}
```

Setelah ada tanda dari proses 1&2, CLK akan dilihat ketika RISING saja (#3). Ketika itu berlangsung, ambil apa saja yang dikirim DIO, lalu ubah ke dalam bentuk desimal per 8 bit data tersebut (#4).

```
if (shbf_done == 1 && clk_pin.state == RISING){
    for (i = 0; i < 8; ++i) {
        dio_dat = bcm2835_gpio_lev( DIO );
        value |= dio_dat << ctr_dio;
```

```
}
```

```
}
```