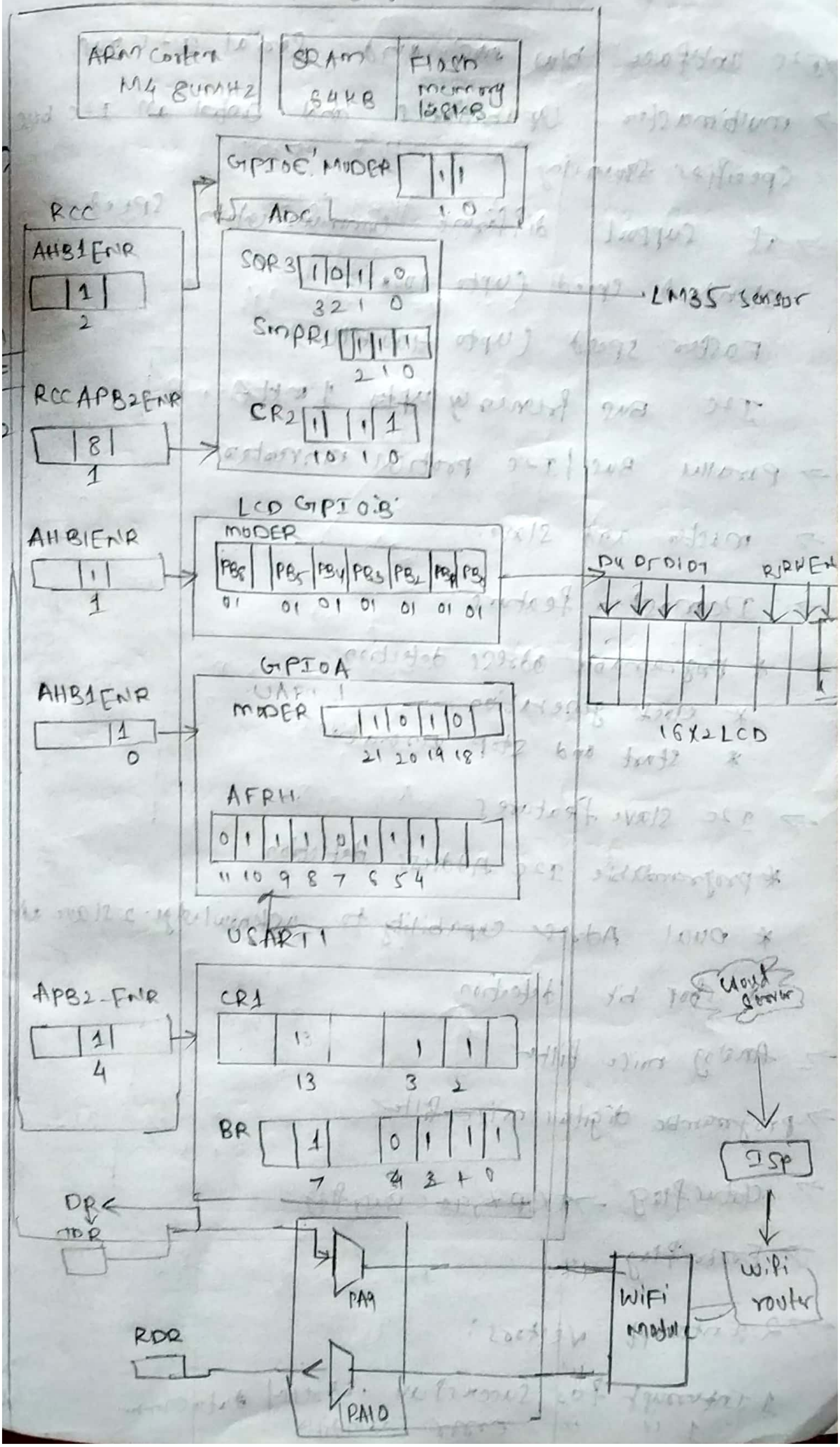
Block diagram for

STM32F401RBT6



Step 2;

pseudo code :-

ADC Initialization ;-

① Enable PORTC clock. RCC-AHB1ENR 2nd bit.

loop:- ② Read 2nd bit if set goto next step. otherwise run loop.

③. Load 11 Bits in GPIOC-MODER in 1st and 0th position to set PC0 Analog mode.

④ Enable ADC clock to set 8th bit in RCC-APB2ENR.

⑤ Load 1010 bits into ADC-SQR3 3,2,1,0 pisition to Enable 10th channel

→ Sensor

Temp	Humidity
27	50

⑥ Load 111 in Bits in ~~Pos~~. ADC_SMPR1 reg pistion 0,1,2 to select 480 cycles

⑦ Load 1 Into ADC-CR2. register in 10, 1, 0 position to enable EOSC, continious mode, A to D conversion ON.

LCD Initialization :-

① Enable PORTB1 clock RCC – AHB1 EnR reg 1st Bit.

② Read RCC – AHB1 EnR register 1st Bit if set goto next step.

③ Load 01 Bits in PB0, PB_1, PB_2, PB3,P PB5, PB8, positions in GPIOB – MODER register.

④ Lcd Initialization ( )
Lcd write – cmd (0X33)

UART. Initialization :-

① Enable PORT 'A' clock RCC – AHB1ENR 0th position.

② Read ~~GPIOA – MODER~~ RCC – AHB1ENR register if set goto next step.

③ Load 10 Bits in 21, 20, 19, 18 to set PA10, PA9 Alternate function mode.

UART config :-

① Enable UART1 Clock RCC – APB2 EnR reg set 4th Bit.

② Load 0111 Bits in 11-4 TO chouse PA9, PA10 Alternate function PORT in GPIOA - AFRH reg.

③ set 13, 3, 2 bits ~~to~~ in USART1_CR1. reg to enable USART, TX, RX.

④ write Baud rate in USART-BRR register.

operation :-

① set 30th Bit in ADC-CR2 reg to regular channel conversion

② Read ADC-SR reg in 2nd Bit Position if set goto next step.

③ read the value in ADC-DR register and assign to global variable. calculate temparature

④ In LCD print string and temparature.

⑤ WiFi command:

⑤ send "AT\r\n" Comand to the USAR1_WiFI. Response_OK() if return value '1' it clear Buffer if '0' return 1.

⑥ "AT+RST\r\n" send to the WiFi_Response if return 1 ie not equal to '0' then goto next step otherwise return '0'.

⑦ "AT+CWMODE = 3\r\n"

if (USART1-WiFi-Response-OK() ==0) goto return 1 otherwise next step., clear Buff

⑧ "AT+CWJAP = \"masvattini\",\"944135727\"";

it will "OK" response. clear Buff

⑨ "AT+CIFSR\r\n" OK response., clear Buff

⑩ "AT+CIPMODE" OK response. clear Buff

⑪ loop: "AT+CIPSTATUS\r\n" response "OK". clear

⑫ "AT+CIPSTART =\"TCP\",\"142.93.218.33\",80\"

responce "OK"; ~~clear~~ ~~go~~

⑬ print the link ."GET/page.php?temp = %d num = 45 & dev = 54 \r\n\r\n", d);

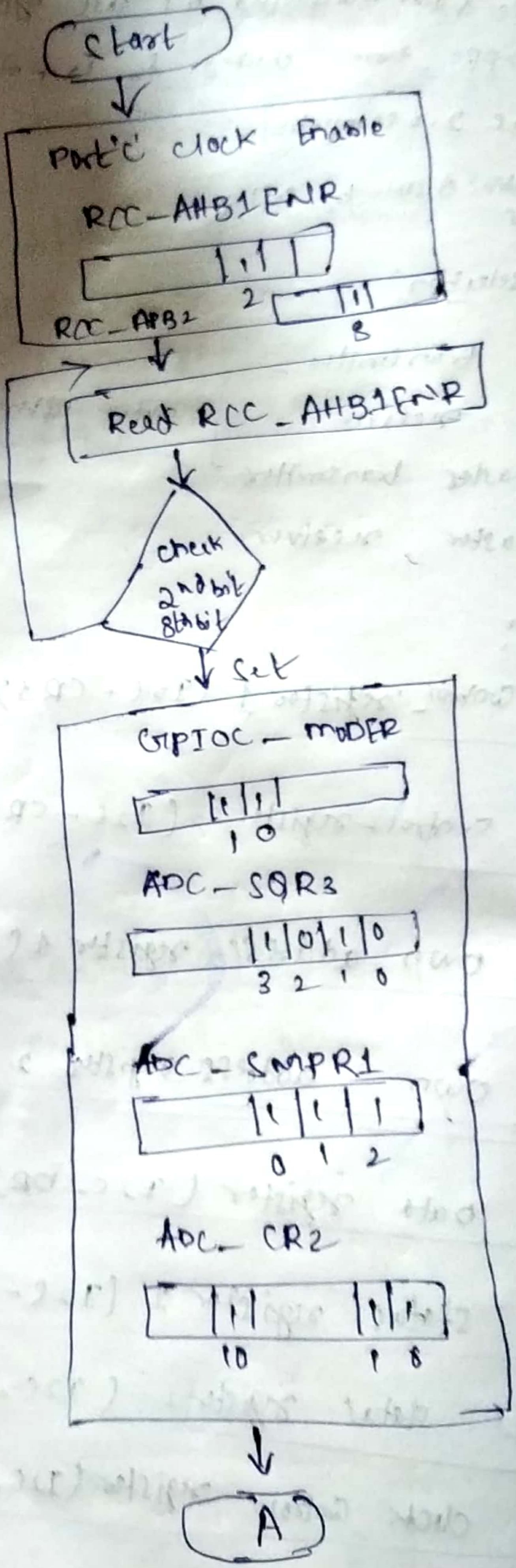
⑭ "AT+CIPSEND = 39\r\n" send number of characters.

⑮ After CIPSEND Rx[i] global Buffer equal to '>' then it will send Fetch data.

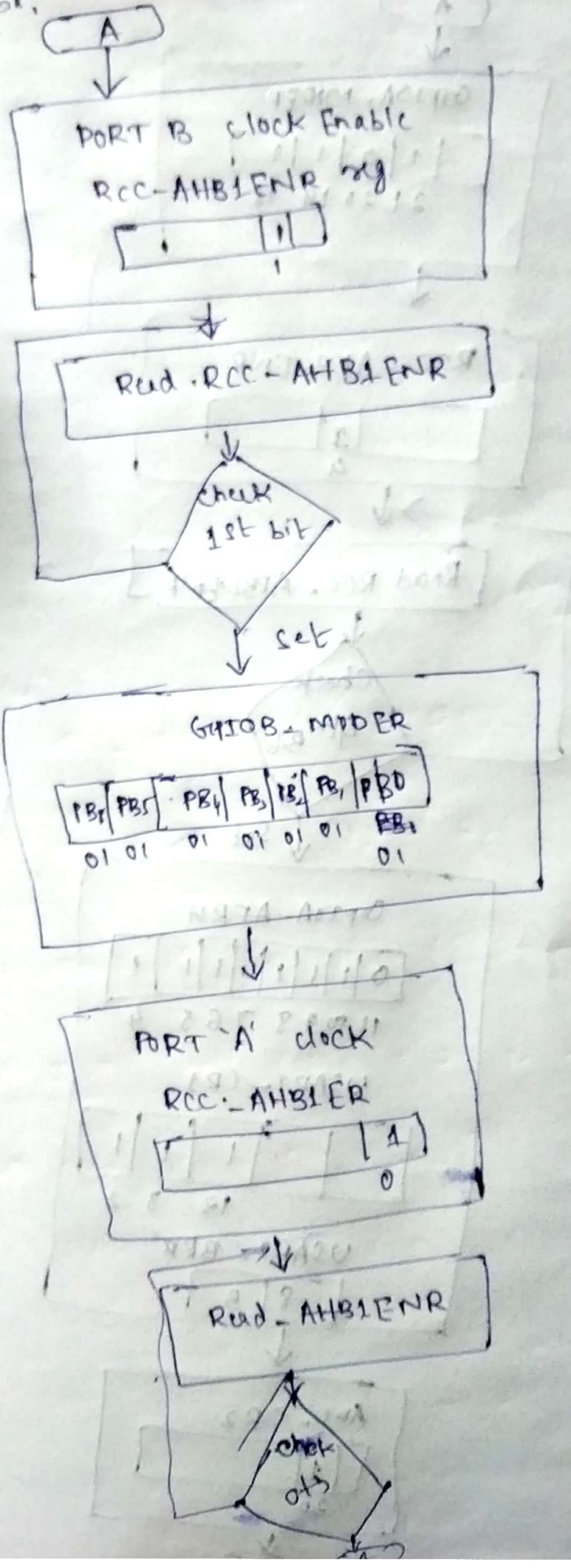
⑯ Send "AT+CIPclose\r\n and maintain delay 5 sec . loop runs.

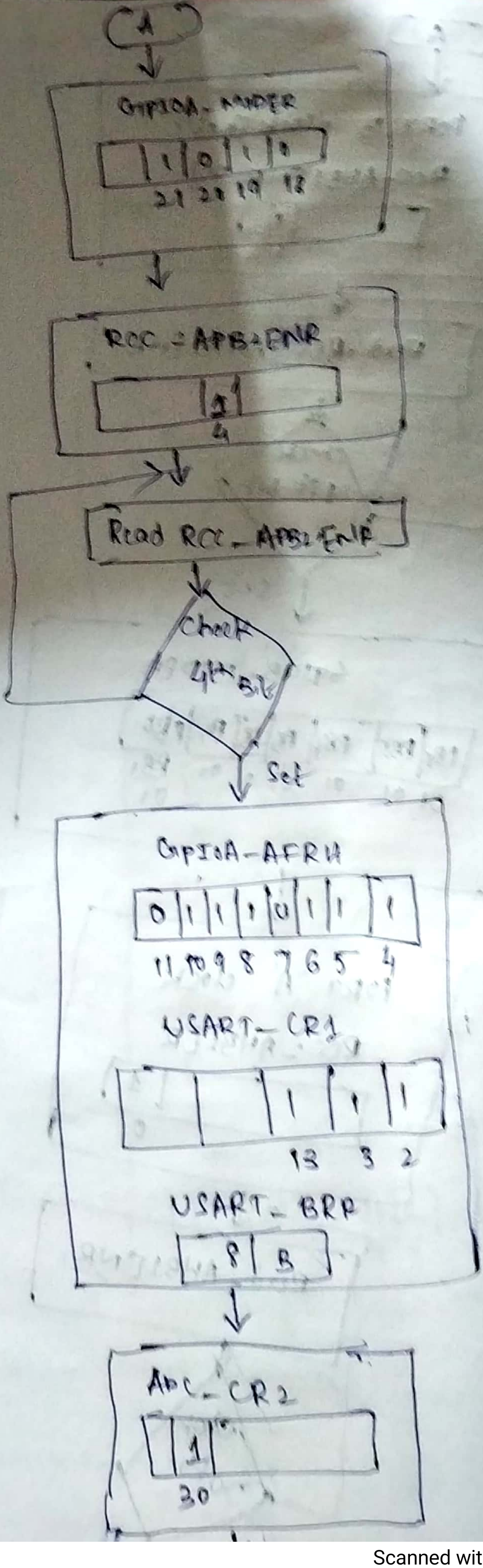
Flow chart:

ADC Initial:

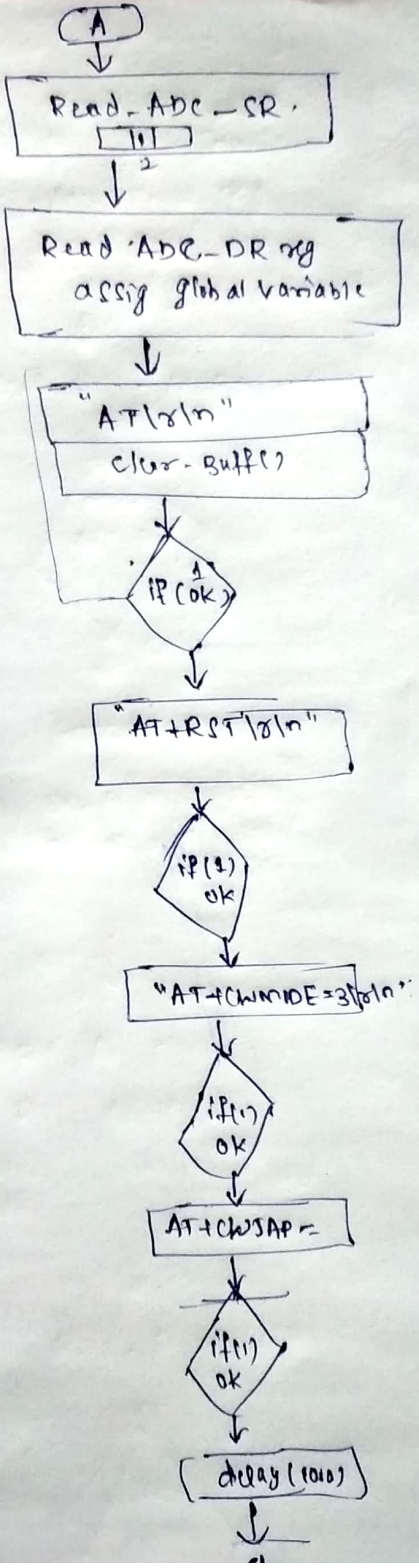


LCD Initi:


A
PORT B clock Enable
RCC-AHB1ENR reg
1
Rud .RCC - AHB1ENR
check 1st bit
set
GPIOB_MODER
PB7 PB5 PB4 PB3 PB2 PB1 PB0
01 01 01 01 01 01 01
PORT 'A' clock
RCC_AHB1ER
1
0
Rud_AHB1ENR
check 0th



A
GPIOA_MODER
1 0 1 0
21 20 19 18
RCC_APB2ENR
1
4
Read RCC_APB2ENR
Check 4th Bit
Set
GPIOA_AFRH
0 1 1 1 0 1 1 1
11 10 9 8 7 6 5 4
USART_CR1
1 1 1
13 3 2
USART_BRR
8 B
ADC_CR2
1
30


A
Read-ADC-SR.
|1|
Read ADC-DR reg
assig global variable
"AT\r\n"
Clear-Buff()
1
if (ok)
"AT+RST\r\n"
if(1)
ok
"AT+CWMODE=3\r\n"
if(1)
ok
AT+CWJAP=
if(1)
ok
delay(1000)