

void interrupt {

num++

if (num == 70) {

led = !led

num = 0;

pinR = 0

Exit from ISR

البرق اضيق دقة التوقيت واحسن

يشرح نظرياً

(1) التوقيت في الدقة دقة يعمل 16K timer

عن طريق المعادلات

timer = timer - 1

$Td \times F_{os}$

$4 \times pres (256 - timer)$

option reg

(2) ما المعادلات المستخدمة؟

في التوقيت في peripheral

(3) timer لا يتم في الدقة بل في 0 فيرجع دقة التوقيت

timer as Counter

الموقع ١٠٠

المسألة: نريد أن نحس الوقت في الكود

الفرق بين timer و Counter

timer → يحس الوقت الداخلي

أو الكود من طرف (D/A)

يستخدم على مصدر خارجي أو تردد

RA1 ← freq

Unsigned int freq;

void main () {

TRISA = 0xFF;

option_Reg = 0x0000;

TMR0 = 0;

تمت التهيئة

do {

delay_ms (1000);

freq = TMR0;

TMR0 = 0;

} while (1);

}

الوقت

توقيت

على

عنوان

لشغل

Counter

التimer 11 (37)

A timer tmr0 is used as counter
the Counter input is connected to
push button (connected to pin RA4) to

so that any button press causes
timer tmr0 to count one pulse

When the number of pulses matches
the number stored in test register

a logic one (5V) appears on pin
PORTC for 3 sec

the voltage activates electro mechanical
relay

الحل

التيار الكهربائي يمر عبر التimer (تعداد 37)

يقوم مصدر خارجي الجانوس (RA4) بزر
الزر

في كل مرة اضغطت الزر واحد

ولو ضغطت عدة مرات الى 37

ويظهر 5V مع التوروس



```
Void main ( ) {
```

```
TRISC = 0 ; portC = 0;
```

```
TRISA = 1;
```

```
option_reg = 0101000;
```

```
TMRA = 0;
```

prescaler = 1

```
while (1) {
```

```
if (TMRA == test)
```

```
{
```

```
portC |= 1; TMRA = 0;
```

```
delay_ms(3000);
```

```
}
```

```
}
```

timers code

timer → oscill

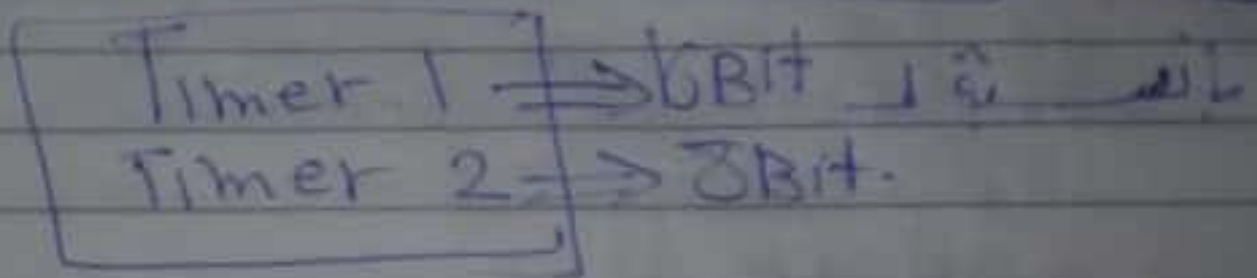
counter → RA4

why wdt?

① على مدار البرنامج - معالجة التي من الممكن ان
ال wdt تمنع حدوث اخطاء انما تعمل Reset للبرنامج

option reg =
PSA = 1

بأنشطة من طرف



المهم في ال

timer ①

capture
compare

هيشنقل ك

timer ②

pres

timer 1 Capture How

الإشارة لتعمل من طرف internal أو external oscillator

يحدث له (Prescaler) ويتحدد هـ يستعمل Falling ولا rising وبعد مدة يدخل CCP1H و CCP1L عن طريق TMRH و TMRH عن طريق TMRH

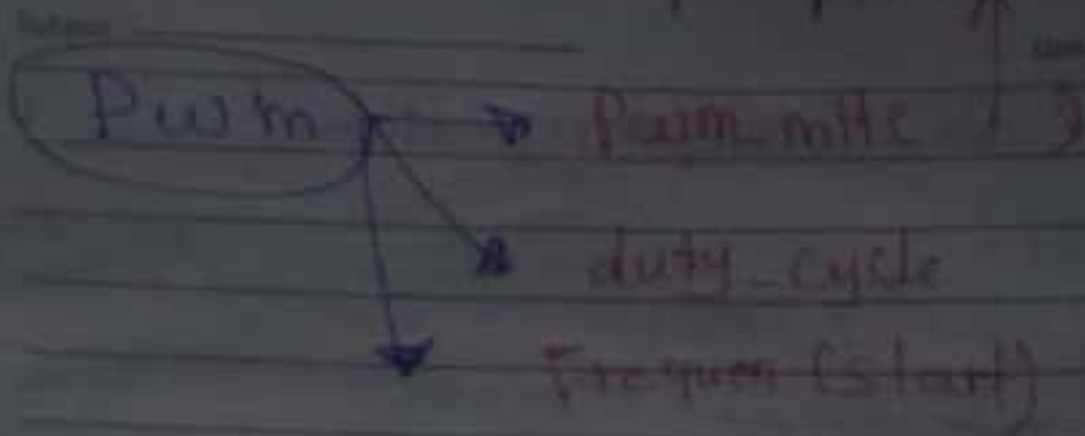


Timer 1 Compare

احتمال كذا في timers ← Counter وله
القياسات مع بطول الإشارة المقابلة مع القيمة
المعطاة مسبقاً

يعني في قيم محددة مسبقاً مثل R2

بعد لحالة له بوصول لقيمة



Example to generate a PWM

on RC2 (Pwm)

Change Duty Cycle from 10 to 100
with step 10

Unsigned int a;

void main() {

PortC = 0;

TRISC = 0; // Frequency

Pwm1 = 0;

Pwm1 = 0;

Pwm2 = 0; set Duty (255)

while(1) {

for (a = 0; a <= 255; a = a + 10)

{ Pwm1 = a; Duty (a)

delay_ms(50);

}

Interrupt

في علم الحاسوب، هو حدث يحدث في المعالج

Polling

دورة امسية
واحدة واحدة
والاستجابة
مباشرة
مفعلة
مفعلة

Interrupt

دورة استجابة سريعة

التم حدوث حدث Interrupt

يخبر المعالج ان الى احدثا هتروية (X)
والتي ان الى احدثا هتروية (stack)
والطلب الى احدثا هتروية او اما يوصل
Interrupt في الوقت (ISR)

2- مراقبة Interrupt لا يكون صغيرة

يمكن ارجع للمكان التي محفوظة في stack
المكان القديم يعني يمكن ارجع لمكان ثاني

خبة لتومين من Interrupt

maskability

لا يمكن ايقافها

non maskable

يمكن ايقافها

يعني في كل وقت منها

ولا يتم ايقافها الى Resource

① INTR

التفعيل الرئيس للـ interrupt

② PIE1

Peripheral ①

③ PIF1

تفعيل مشترك

④ PIE2

ممكن التفعيل بواسطة Per

⑤ PIE2

Peripheral ①

PK ②

لشأن التشغيل أي interrupt في Global يكون
Flag يكون وضعه في

برنامج

أما على مستوى Seven Segment

لا يحتاج إلى interrupt ولا سؤال

unsigned int Count = 0;
unsigned Segment[] = {

0b000, 0b001, 0b0010
};

void main() {

ADCON1 = 0x06;

TRISB = 0xFF;

TRISD = 0; PORTD = 0;

TRISE = 0; PORTE = 0;

TRISC0bit = 0;

OPTION_REG.INTEDG = 0;

INTCON.GIE = 1;

INTCON.INTE = 1;

PORTC.RC0 = 1;

PORTD = Segment[mRC0];

while(1) {

digital++

PORTD = Segment[int[digital]];

delay_ms(200); digital++;

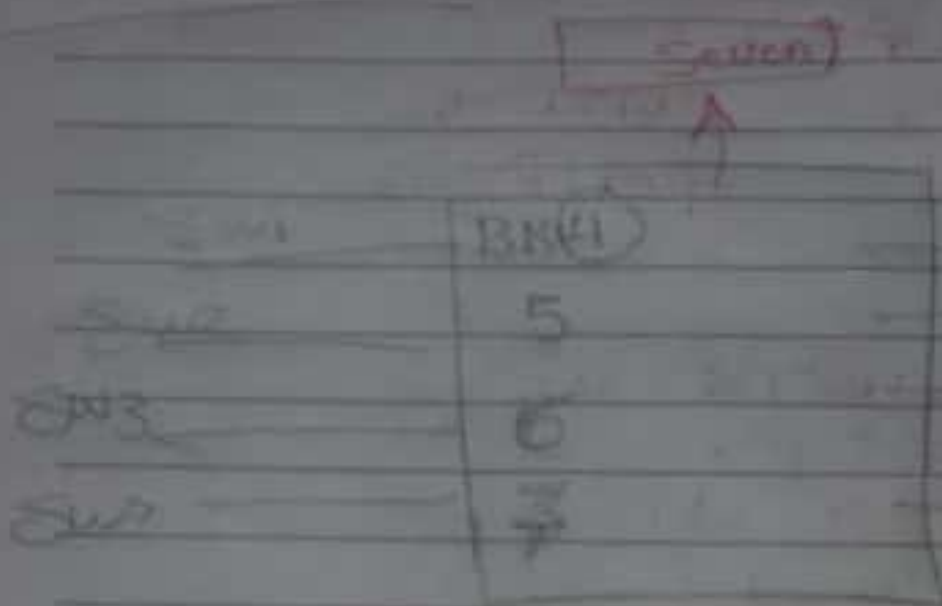
if(digital == 10) digital = 0;

}

void main

```
{  
  intcon, TMR0 = 0;  
  portc.RC0 = 0;  
  Delay_ms(1000);  
  portc.RC0 = 1;  
}
```

243- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- 16- 17- 18- 19- 20- 21- 22- 23- 24- 25- 26- 27- 28- 29- 30- 31- 32- 33- 34- 35- 36- 37- 38- 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74- 75- 76- 77- 78- 79- 80- 81- 82- 83- 84- 85- 86- 87- 88- 89- 90- 91- 92- 93- 94- 95- 96- 97- 98- 99- 100- 101- 102- 103- 104- 105- 106- 107- 108- 109- 110- 111- 112- 113- 114- 115- 116- 117- 118- 119- 120- 121- 122- 123- 124- 125- 126- 127- 128- 129- 130- 131- 132- 133- 134- 135- 136- 137- 138- 139- 140- 141- 142- 143- 144- 145- 146- 147- 148- 149- 150- 151- 152- 153- 154- 155- 156- 157- 158- 159- 160- 161- 162- 163- 164- 165- 166- 167- 168- 169- 170- 171- 172- 173- 174- 175- 176- 177- 178- 179- 180- 181- 182- 183- 184- 185- 186- 187- 188- 189- 190- 191- 192- 193- 194- 195- 196- 197- 198- 199- 200- 201- 202- 203- 204- 205- 206- 207- 208- 209- 210- 211- 212- 213- 214- 215- 216- 217- 218- 219- 220- 221- 222- 223- 224- 225- 226- 227- 228- 229- 230- 231- 232- 233- 234- 235- 236- 237- 238- 239- 240- 241- 242- 243- 244- 245- 246- 247- 248- 249- 250- 251- 252- 253- 254- 255- 256- 257- 258- 259- 260- 261- 262- 263- 264- 265- 266- 267- 268- 269- 270- 271- 272- 273- 274- 275- 276- 277- 278- 279- 280- 281- 282- 283- 284- 285- 286- 287- 288- 289- 290- 291- 292- 293- 294- 295- 296- 297- 298- 299- 300- 301- 302- 303- 304- 305- 306- 307- 308- 309- 310- 311- 312- 313- 314- 315- 316- 317- 318- 319- 320- 321- 322- 323- 324- 325- 326- 327- 328- 329- 330- 331- 332- 333- 334- 335- 336- 337- 338- 339- 340- 341- 342- 343- 344- 345- 346- 347- 348- 349- 350- 351- 352- 353- 354- 355- 356- 357- 358- 359- 360- 361- 362- 363- 364- 365- 366- 367- 368- 369- 370- 371- 372- 373- 374- 375- 376- 377- 378- 379- 380- 381- 382- 383- 384- 385- 386- 387- 388- 389- 390- 391- 392- 393- 394- 395- 396- 397- 398- 399- 400- 401- 402- 403- 404- 405- 406- 407- 408- 409- 410- 411- 412- 413- 414- 415- 416- 417- 418- 419- 420- 421- 422- 423- 424- 425- 426- 427- 428- 429- 430- 431- 432- 433- 434- 435- 436- 437- 438- 439- 440- 441- 442- 443- 444- 445- 446- 447- 448- 449- 450- 451- 452- 453- 454- 455- 456- 457- 458- 459- 460- 461- 462- 463- 464- 465- 466- 467- 468- 469- 470- 471- 472- 473- 474- 475- 476- 477- 478- 479- 480- 481- 482- 483- 484- 485- 486- 487- 488- 489- 490- 491- 492- 493- 494- 495- 496- 497- 498- 499- 500- 501- 502- 503- 504- 505- 506- 507- 508- 509- 510- 511- 512- 513- 514- 515- 516- 517- 518- 519- 520- 521- 522- 523- 524- 525- 526- 527- 528- 529- 530- 531- 532- 533- 534- 535- 536- 537- 538- 539- 540- 541- 542- 543- 544- 545- 546- 547- 548- 549- 550- 551- 552- 553- 554- 555- 556- 557- 558- 559- 560- 561- 562- 563- 564- 565- 566- 567- 568- 569- 570- 571- 572- 573- 574- 575- 576- 577- 578- 579- 580- 581- 582- 583- 584- 585- 586- 587- 588- 589- 590- 591- 592- 593- 594- 595- 596- 597- 598- 599- 600- 601- 602- 603- 604- 605- 606- 607- 608- 609- 610- 611- 612- 613- 614- 615- 616- 617- 618- 619- 620- 621- 622- 623- 624- 625- 626- 627- 628- 629- 630- 631- 632- 633- 634- 635- 636- 637- 638- 639- 640- 641- 642- 643- 644- 645- 646- 647- 648- 649- 650- 651- 652- 653- 654- 655- 656- 657- 658- 659- 660- 661- 662- 663- 664- 665- 666- 667- 668- 669- 670- 671- 672- 673- 674- 675- 676- 677- 678- 679- 680- 681- 682- 683- 684- 685- 686- 687- 688- 689- 690- 691- 692- 693- 694- 695- 696- 697- 698- 699- 700- 701- 702- 703- 704- 705- 706- 707- 708- 709- 710- 711- 712- 713- 714- 715- 716- 717- 718- 719- 720- 721- 722- 723- 724- 725- 726- 727- 728- 729- 730- 731- 732- 733- 734- 735- 736- 737- 738- 739- 740- 741- 742- 743- 744- 745- 746- 747- 748- 749- 750- 751- 752- 753- 754- 755- 756- 757- 758- 759- 760- 761- 762- 763- 764- 765- 766- 767- 768- 769- 770- 771- 772- 773- 774- 775- 776- 777- 778- 779- 780- 781- 782- 783- 784- 785- 786- 787- 788- 789- 790- 791- 792- 793- 794- 795- 796- 797- 798- 799- 800- 801- 802- 803- 804- 805- 806- 807- 808- 809- 810- 811- 812- 813- 814- 815- 816- 817- 818- 819- 820- 821- 822- 823- 824- 825- 826- 827- 828- 829- 830- 831- 832- 833- 834- 835- 836- 837- 838- 839- 840- 841- 842- 843- 844- 845- 846- 847- 848- 849- 850- 851- 852- 853- 854- 855- 856- 857- 858- 859- 860- 861- 862- 863- 864- 865- 866- 867- 868- 869- 870- 871- 872- 873- 874- 875- 876- 877- 878- 879- 880- 881- 882- 883- 884- 885- 886- 887- 888- 889- 890- 891- 892- 893- 894- 895- 896- 897- 898- 899- 900- 901- 902- 903- 904- 905- 906- 907- 908- 909- 910- 911- 912- 913- 914- 915- 916- 917- 918- 919- 920- 921- 922- 923- 924- 925- 926- 927- 928- 929- 930- 931- 932- 933- 934- 935- 936- 937- 938- 939- 940- 941- 942- 943- 944- 945- 946- 947- 948- 949- 950- 951- 952- 953- 954- 955- 956- 957- 958- 959- 960- 961- 962- 963- 964- 965- 966- 967- 968- 969- 970- 971- 972- 973- 974- 975- 976- 977- 978- 979- 980- 981- 982- 983- 984- 985- 986- 987- 988- 989- 990- 991- 992- 993- 994- 995- 996- 997- 998- 999- 1000



4 5 6 7

4 5 6 7

60 V analog VITC
 - > +
 (1) Logic level
 analog

(2) \rightarrow

400 MHz

100 MHz

50 MHz

25 MHz

12.5 MHz

6.25 MHz

3.125 MHz

1.5625 MHz

781.25 kHz

390.625 kHz

195.3125 kHz

97.65625 kHz

48.828125 kHz

24.4140625 kHz

12.20703125 kHz

6.103515625 kHz

3.0517578125 kHz

1.52587890625 kHz

762.939453125 Hz

381.4697265625 Hz

190.73486328125 Hz

95.367431640625 Hz

47.6837158203125 Hz

23.84185791015625 Hz

11.920928955078125 Hz

5.9604644775390625 Hz

2.98023223876953125 Hz

1.4901161193847656 Hz

0.7450580596923828 Hz

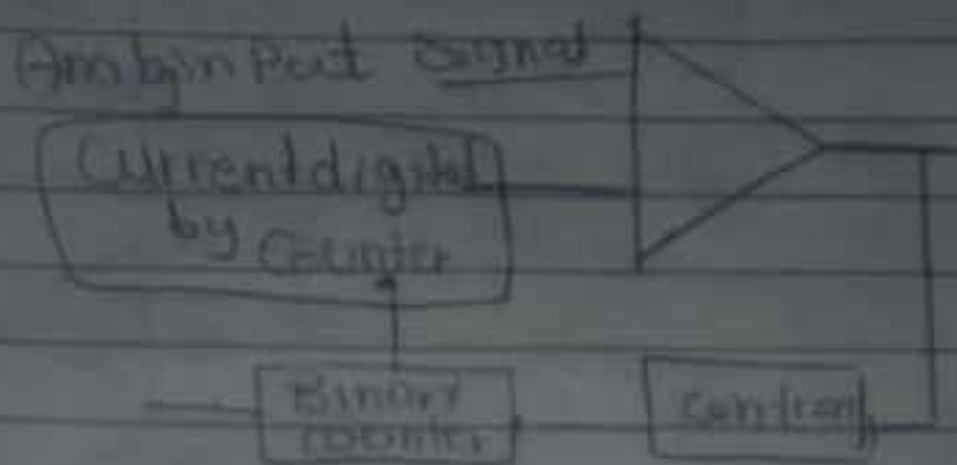
0.3725290298461914 Hz

(2)

Final



Tracking Converter



1. depend

* Comparator

* Counter Analog

* Digital to Converter

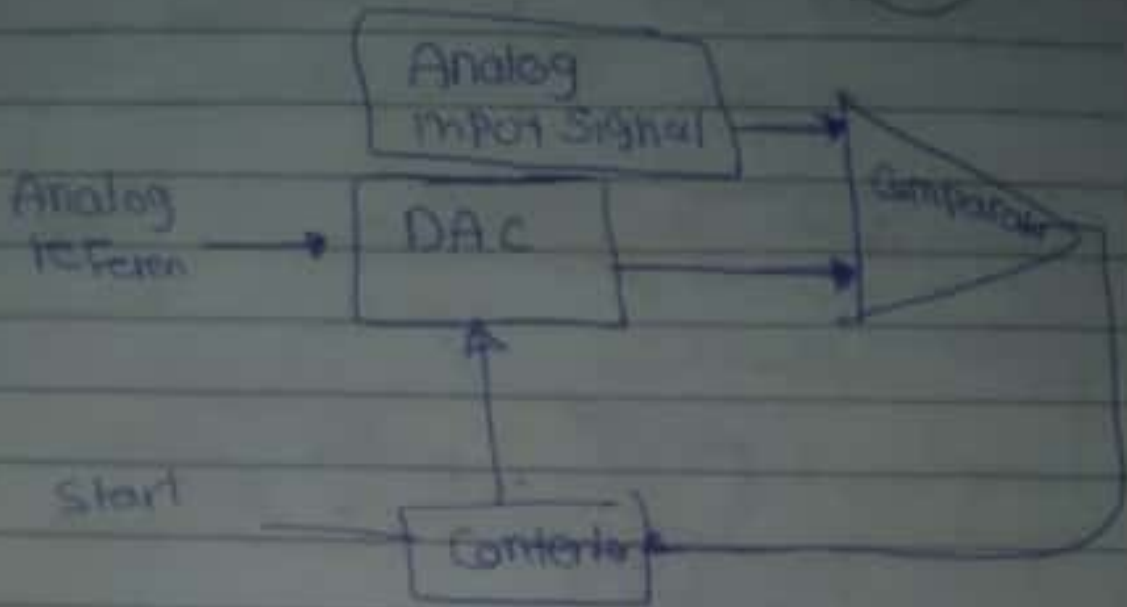
القلب الإلكتروني Counter وهو دائرة رقم إلى رقم Analog إلى Digital Converter وظيفته تحويل القيمة Analog إلى Digital وهو موجود في الطرف الثاني من Converter. ولذا إذا أردنا أن نخرج من Counter وإغلاق دائرة التحويل ونسعى لحدوثها إن شاء الله

بعض حيل ما نستخدمه في الأنظمة السريعة

(3)

Successive

Counter Binary Search



Register Binary Search
 Digital set most Significant

Comparator most clear Reset

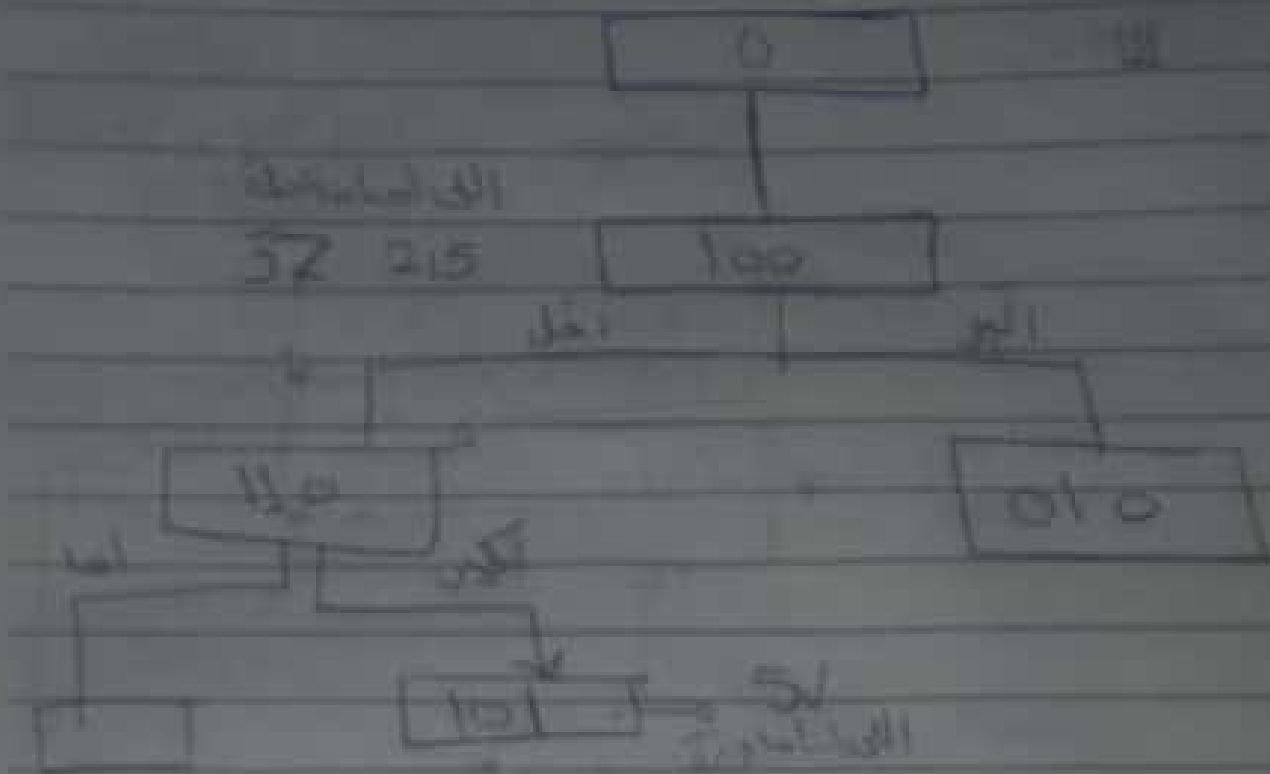
النتيجة

tracking Counter
 Successive Binary

4

الم

المادة 37 حول (أ) ذلك القصة التي داخلها
 القصة النهائية 3



Signal Conditioning

لما القصة التي كانت في (5-5)



معدل إشارة 2.5V

0-5

1000

5
1000

2.5

2.5 = 5

5

العدد

A/D		نتیجہ حاصل	
A/D	Result	High	Register
A/D	Result	Low	Register
A/D	Result	High	Register
A/D	Result	Low	Register

High & Low 16 Bit کے مقادیر

لیکن ہمیں یہ معلوم کرنا ہے کہ یہ کتنے
 کیونکہ یہ 16 Bit کے مقادیر ہیں
 لیکن یہ 16 Bit کے مقادیر ہیں
 لیکن یہ 16 Bit کے مقادیر ہیں

لیکن یہ 16 Bit کے مقادیر ہیں
 لیکن یہ 16 Bit کے مقادیر ہیں

لیکن یہ 16 Bit کے مقادیر ہیں

لیکن یہ 16 Bit کے مقادیر ہیں

6

6

البرامج

محتاجين دالين عشان تعامل

ADC

ADC_Init();

tmp = ADC_Read(0);

اسم
المتغير

char str[67]

Assigned int temp;

float v;

void main()

TrisA = 0x00000000;

ADC_Init();

led_init();

led_read_led_cursor_off();

for (i = 0;

i

temp = ADC_Read(2)

v = temp * 5
1023

float to str

7

lcd_out(1,1, 'Value');
delay_ms(1000);

lcd_out_CP(SH);
delay_ms(100);

g

لیکچر میں غلطی سے
to SH
پیشکش لکھ کر لکھ کر

SH 81

81

Timero 1 a 0.000

Example: Flash a LED connected to RCO with
toF = 4 * prescaler * 256 - TMR0 = 13,1 ms
Fore

II = delay 10,11
- 10,11

Solution

the delay (over flow) is the number of

the of over flow = $\frac{td}{\tau_{00}}$

$$no = \text{over flow} = \frac{1 \times 10^6}{4 \times 256 \times (256 - 0)} = 76$$

(76) = 1,115 interrupt, so 10,11

1- unsigned int num;

2- void main() {

3- num = 0;

4- TMR0 = 0;

5- OPTION_REG = 0x00;

6- TMR0 = 0; initial

7- INTCON = 0b10100000; enable General
enable timer

8- do {

}