

Micro-Controller Experiment

Week7

Teacher: 廖裕評 Yu-Ping Liao

TA: 陳大荃 Da-chuan Chen, 陳恩妮 En-ni Chen

Class Rules

1. No drink besides water.
2. Bring a laptop and breadboard if needed.
3. Ask us TAs to sign and borrow development boards. Do not sign or ask others to sign for you without TAs' permission.
4. Arriving 10 minutes after the bell rings will be regarded as absent.
5. If you damage any borrowed equipment, you have to pay for it.

Homework Rules

1. Includes: A. Class content, B. Class exercise, C. Homework (screenshot or video)
2. Editing software: MS PowerPoint
3. File format: PDF
4. Filename: "date_group_studentID_name.pdf", like "0916_第1組_11028XXX_陳OO.pdf"
5. The homework deadline is 23:59 of the day before the next class. If you are late, then your grade will be deducted.

Contact

If you encounter any problems with this class, please get in touch with us with the following E-mails:

1. Teacher, Prof. Yu-Ping Liao 廖裕評 : lyp@cycu.org.tw
2. TA, Da-chuan Chen 陳大荃 : dachuan516@gmail.com
3. TA, En-ni Chen 陳恩妮 : anna7125867@gmail.com

Or visit 篤信 Lab353 for further questions.

Outline of the Week

1. PWM introduction
2. PWM Project.
3. Homework 7-1.
4. Homework 7-2.
5. Homework 7-3.
6. Homework 7-4 Bonus.

The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. A large, solid red speech bubble is centered on the page, pointing downwards.

PWM Introduction

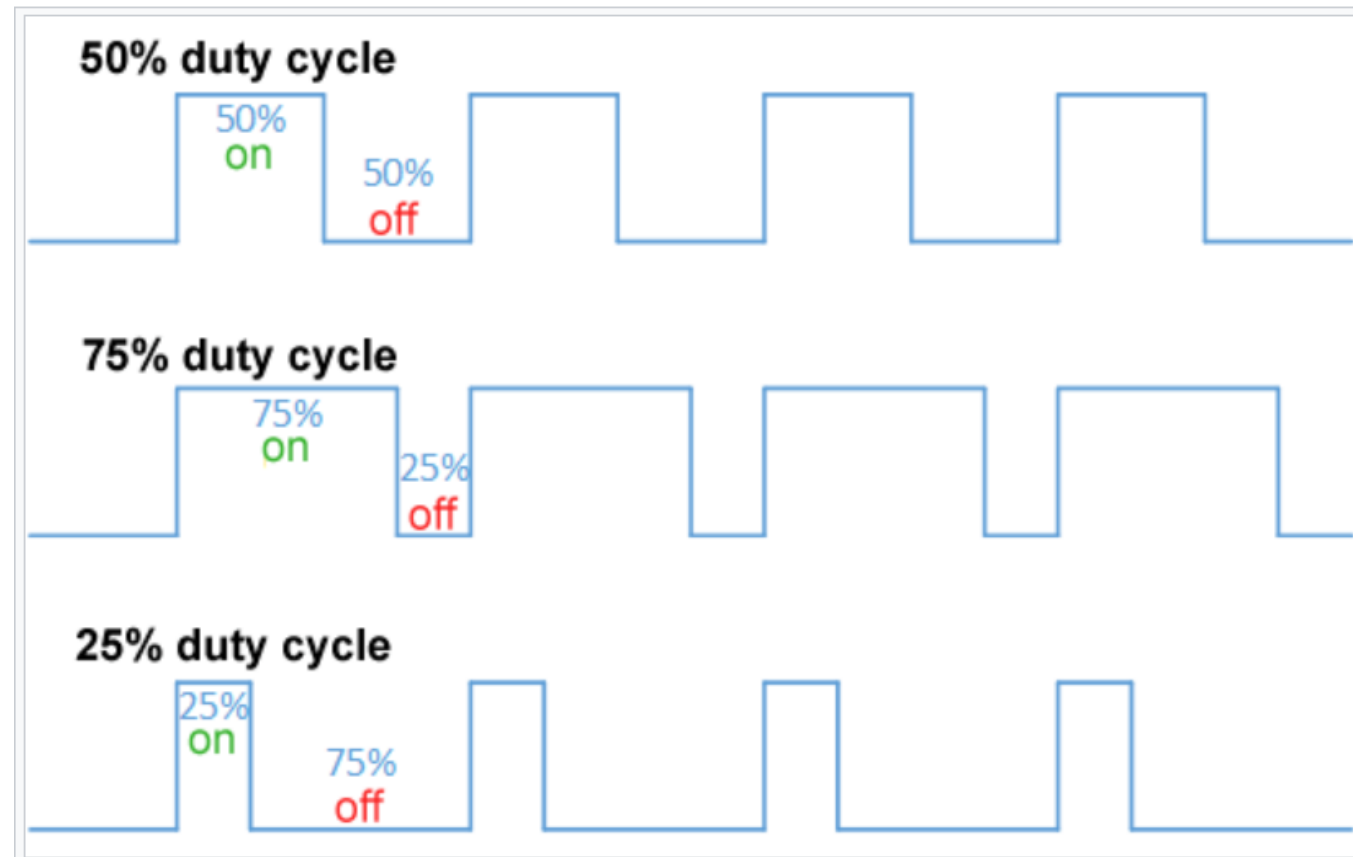
PWM

Pulse-width modulation (PWM) is a technique for generating analog signals using pulse waves. Typically, the converted pulse wave has a fixed period, but the duty cycle of the pulse wave varies depending on the magnitude of the analog signal.

The **duty cycle** and **frequency** of a PWM signal determine its behavior.

Duty Cycle

The term duty cycle describes the proportion of on time to the regular interval or period of time.



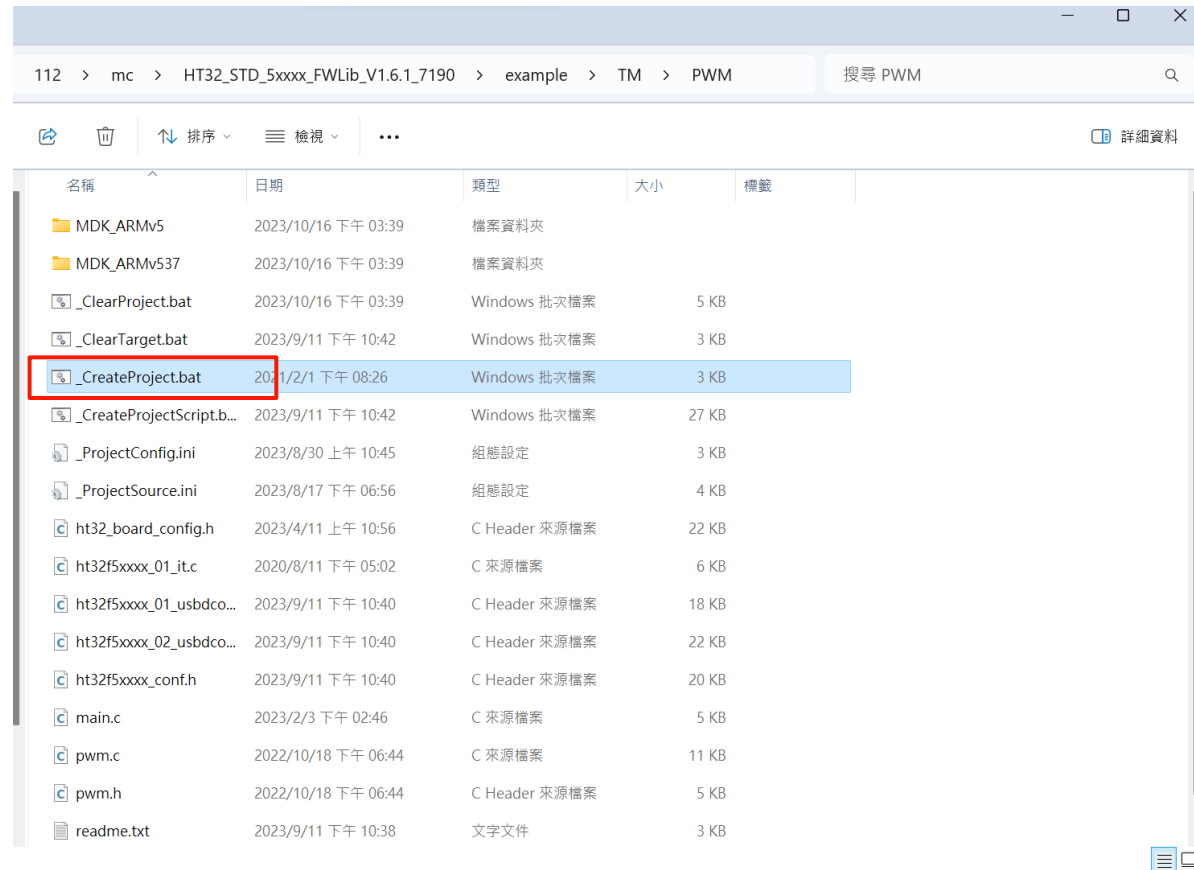
The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. A large red speech bubble is centered on the page, pointing downwards.

PWM Project

1. Execute “_CreatProject”

1. Go to “~/HT32_STD_5xxxx_FWLib_V1.5.1_7084/example/TM/PWM”.

2. Double click “_CreateProject.bat”.



2. Launch project

Figure 2 illustrates the process of launching a project in the μVision IDE. The left panel shows a file explorer view of the project directory, highlighting the **MDK_ARMv537** folder and the **Project_52352.uvprojx** file. The right panel shows the μVision IDE interface with the **Project** window displaying the project structure and the **main.c** file open in the editor. The **Build Output** window at the bottom shows the compilation command and the result: **0 Error(s), 0 Warning(s)**.

File Explorer View:

名稱	日期
MDK_ARMv5	2023/9/26 下午 10:40
MDK_ARMv537	2023/9/26 下午 10:40
_ClearProject.bat	2023/9/26 下午 10:40
_ClearTarget.bat	2023/9/11 下午 10:40
_CreateProject.bat	2021/2/1 下午 10:40
_CreateProjectScript.b...	2023/9/11 下午 10:40
_ProjectConfig.bat	2022/5/26 上午 10:40
_ProjectConfig.ini	2022/5/26 上午 10:40
ht32_board_config.h	2022/4/12 下午 10:40
ht32f5xxx_01_it.c	2018/8/3 下午 10:40
ht32f5xxx_01_usbdco...	2023/9/11 下午 10:40
ht32f5xxx_02_usbdco...	2023/9/11 下午 10:40
ht32f5xxx_conf.h	2023/9/11 下午 10:40
i2cm.h	2016/5/25 下午 02:39
main.c	2020/8/26 下午 07:06

μVision IDE Interface:

Project Window:

- Project: Project_52352
- HT32
 - User
 - main.c
 - ht32f5xxx
 - pwm.c
 - Config
 - CMSIS
 - MDK-ARM
 - USBD_Library
 - Library
 - Retarget
 - Utilities
 - Doc

main.c File Content:

```
1  /*
2  * @file    TM/PWM/main.c
3  * @version $Rev:: 6712 $
4  * @date    $Date:: 2023-02-03 # $
5  * @brief    Main program.
6  *
7  * @attention
8  *
9  * Firmware Disclaimer Information
10 *
11 * 1. The customer hereby acknowledges and agrees that the program technical documentation, including the
12 * code, which is supplied by Holtek Semiconductor Inc., (hereinafter referred to as "HOLTEK") is the
13 * proprietary and confidential intellectual property of HOLTEK, and is protected by copyright law and
14 * other intellectual property laws.
15 *
16 * 2. The customer hereby acknowledges and agrees that the program technical documentation, including the
17 * code, is confidential information belonging to HOLTEK, and must not be disclosed to any third parties
18 * other than HOLTEK and the customer.
19 *
20 * 3. The program technical documentation, including the code, is provided "as is" and for customer reference
21 * only. After delivery by HOLTEK, the customer shall use the program technical documentation, including
22 * the code, at their own risk. HOLTEK disclaims any expressed, implied or statutory warranties, including
23 * the warranties of merchantability, satisfactory quality and fitness for a particular purpose.
24 *
25 * <h2><center>Copyright (C) Holtek Semiconductor Inc. All rights reserved</center></h2>
26 *
27 *
28 * Includes -----*/
```

Build Output Window:

```
After Build - User command #1: fromelf --bin -o ".\HT32\52352\Obj\HT32.axf.bin" ".\HT32\52352\Obj\HT32.axf"
".\HT32\52352\Obj\HT32.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:11
```

main

```
54 int main(void)
55 {
56     PWM_Init();
57
58     PWM_UpdateDuty(PWM_CH0, PWM_DUTY_50);
59     PWM_UpdateDuty(PWM_CH1, PWM_DUTY_25);
60     PWM_Cmd(ENABLE);
61
62     Delay(5000);
63
64     PWM_UpdateDuty(PWM_CH0, PWM_DUTY_0);
65     PWM_UpdateDuty(PWM_CH1, PWM_DUTY_75);
66
67     Delay(5000);
68
69     PWM_Cmd(DISABLE);
70
71     Delay(5000);
72
73     PWM_SetFreq(PWM_FREQ_12K);
74     PWM_UpdateDuty(PWM_CH0, PWM_FREQ_12K * 0.25);
75     PWM_UpdateDuty(PWM_CH1, PWM_FREQ_12K * 0.75);
76     PWM_Cmd(ENABLE);
77
78     while (1);
79 }
```

PWM Initialization Setting

PWM Enable

Set Duty Cycle as X%

PWM Disable

Set CRR value

F12"PWM_FREQ_12K"->PWM.h

```

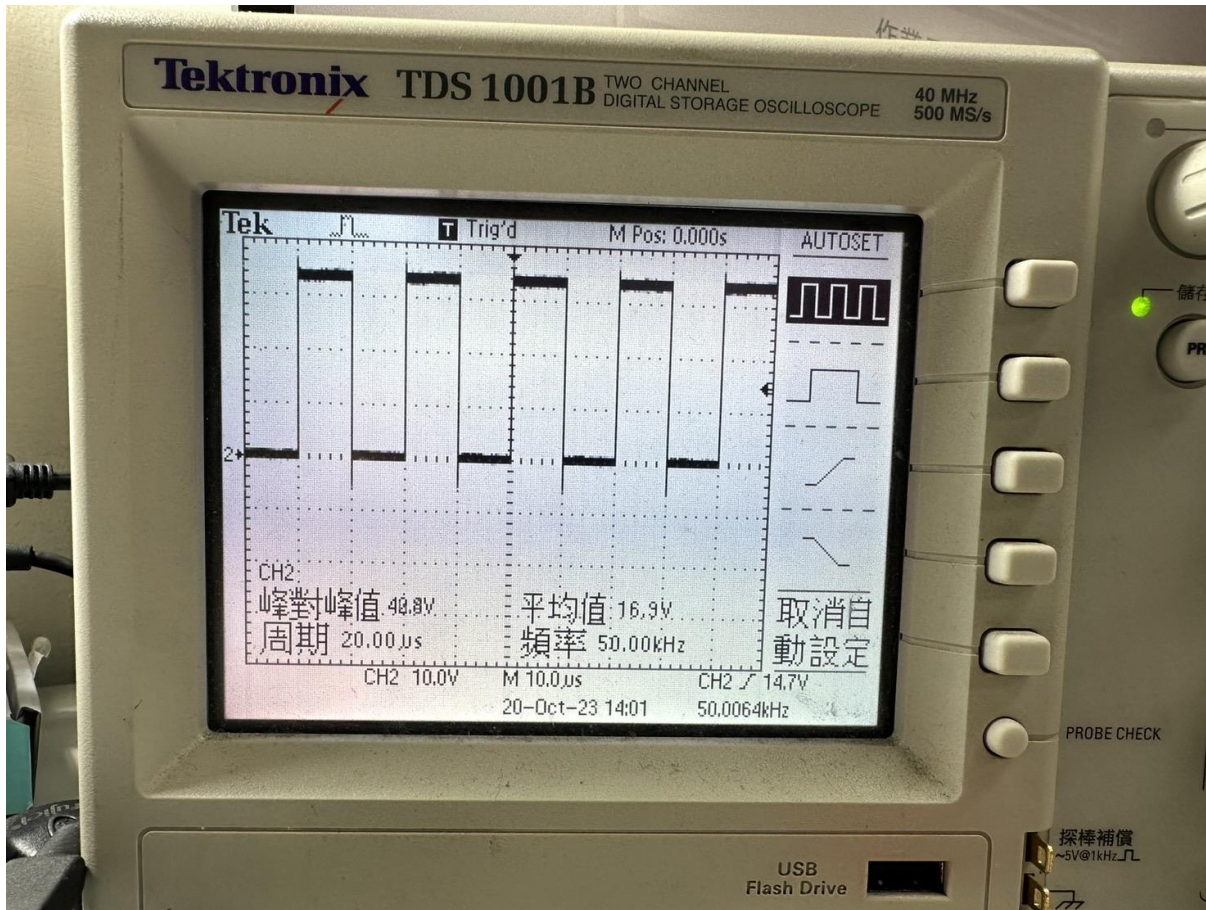
77 #define PWM_DUTY_0      (0)
78 #define PWM_DUTY_25     (HTCFG_PWM_TM_RELOAD * 0.25)
79 #define PWM_DUTY_50     (HTCFG_PWM_TM_RELOAD * 0.5)
80 #define PWM_DUTY_75     (HTCFG_PWM_TM_RELOAD * 0.75)
81 #define PWM_DUTY_100    (0xFFFF)
82
83 #define PWM_FREQ_50K     (HTCFG_PWM_TM_PCLK / HTCFG_PWM_TM_PRESCALER / 50000)
84 #define PWM_FREQ_40K     (HTCFG_PWM_TM_PCLK / HTCFG_PWM_TM_PRESCALER / 40000)
85 #define PWM_FREQ_12K     (HTCFG_PWM_TM_PCLK / HTCFG_PWM_TM_PRESCALER / 12000)
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

```

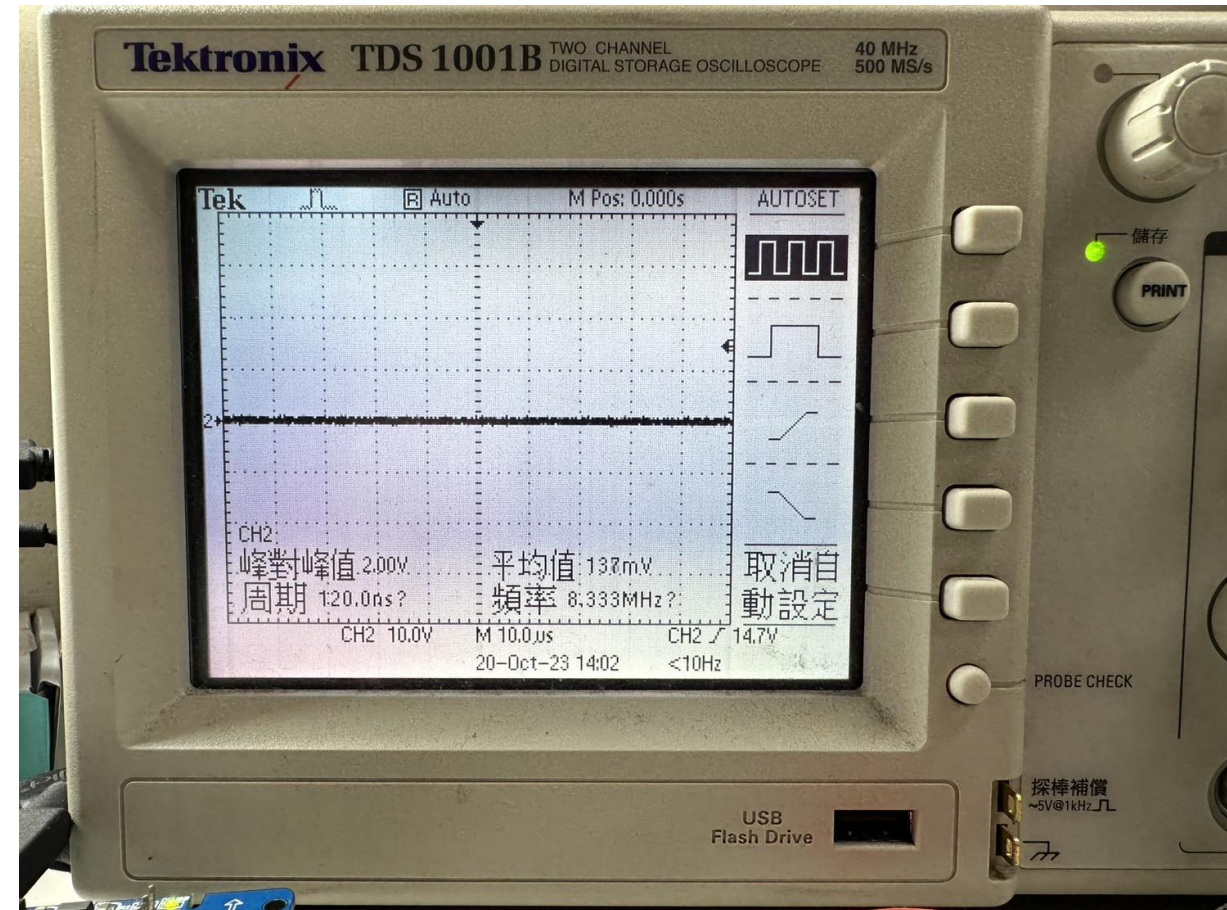
$$\text{TM_RELOAD} = \text{CRR} = 48000000 / 1 / 50000 = 960 < 65535$$
$$T = 1/f = 1/50000 = 0.0002\text{s} = 20\mu\text{s}$$

PS: Since GPTM is a 16-bit up/down counter, if CRR exceeds 16 bits, it cannot be displayed correctly.

Show on the oscilloscope

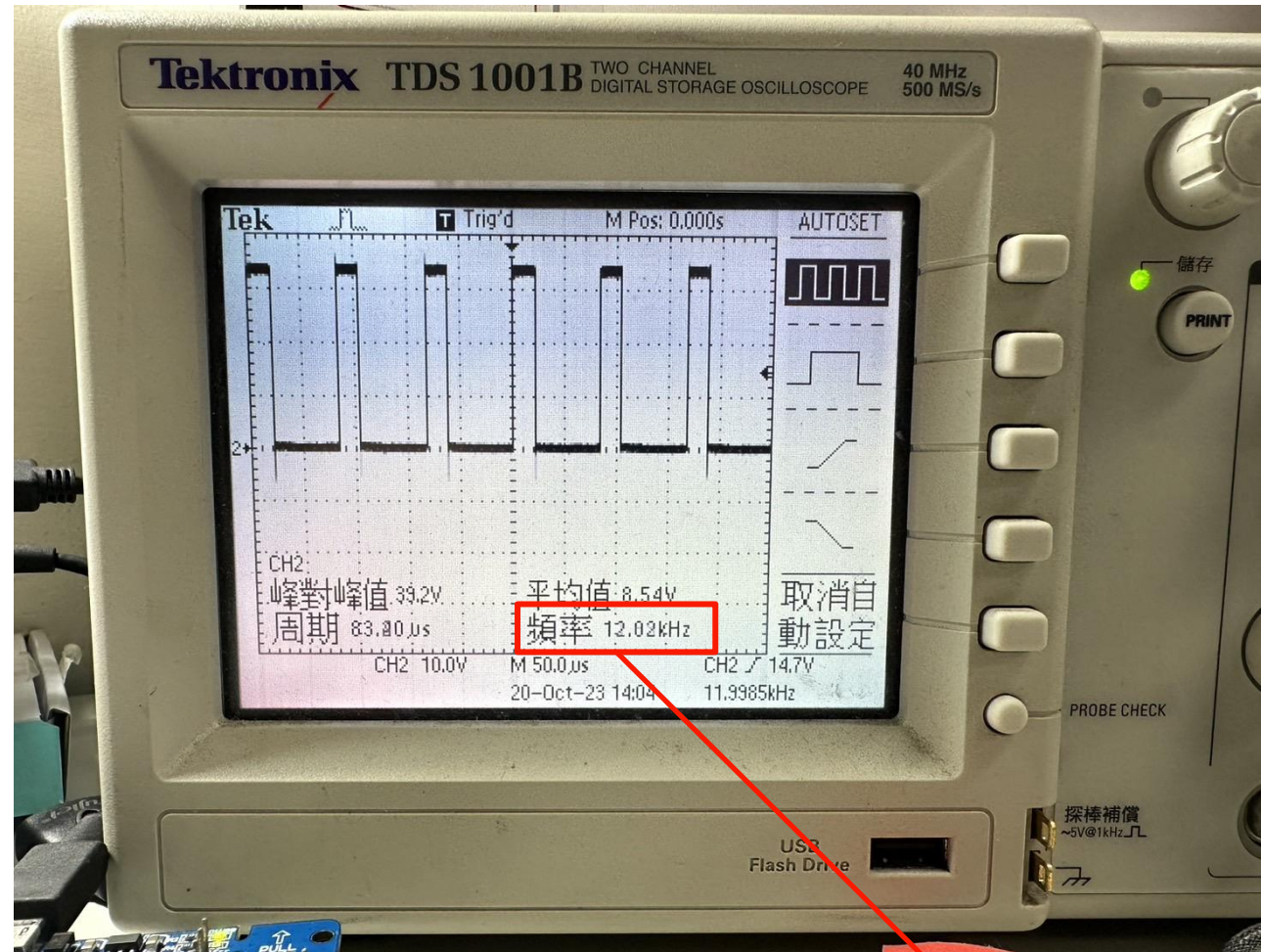


➤ PWM_UpdateDuty(PWM_CH0, PWM_DUTY_50);



➤ PWM_UpdateDuty(PWM_CH0, PWM_DUTY_0);

Show on the oscilloscope



➤ PWM_UpdateDuty(PWM_CH0, PWM_FREQ_12K * 0.25);

Homework W7-1.

[https://github.com/CYCU-AIoT-System-
Lab/Microcontroller-
Experiment/blob/main/w7/TM-PWM-
Experiment_Steps.md](https://github.com/CYCU-AIoT-System-Lab/Microcontroller-Experiment/blob/main/w7/TM-PWM-Experiment_Steps.md)

Execute the example and show on oscilloscope

- Objective: Observe output signal with oscilloscope and explain the code.
- Hint:
 1. Use key "F12" to locate specified pins and connect them.
 2. Increase the default delay time for result observation.
 3. Connect PWM **CH0** and **CH1** to the oscilloscope.

☆ PS. Please record.



The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. A large red speech bubble is centered on the page, containing the text 'Homework W7-2'.

Homework W7-2

Control servo motor (SG90)

- Objective: Modify code and connect **CH0** to oscilloscope, and use **CH1** to control SG90.
- Hint:
 1. Connect **CH0** to oscilloscope.
 2. Connect **CH1** to servo motor SG90, and rotate it to 0, 45, 90, 135, and 180 degrees angle.
 3. Edit code in files: main.c & PWM.h.

☆ PS. Please record.

SG90

Datasheet: <https://reurl.cc/6DX3pV>

Feature:

```
38  /* Settings -----*/
39  #define HTCFG_PWM_TM_PRESCALER          // 1 ~ 65535
40  #define HTCFG_PWM_FREQ_HZ              (50)
41  #define HTCFG_PWM_IDLE_STATE           (0) // 0: 0 duty, 1: 100 duty
42
43  #define HTCFG_PWM_TM_RELOAD             (HTCFG_PWM_TM_PCLK / HTCFG_PWM_TM_PRESCALER / HTCFG_PWM_FREQ_HZ)
44  #if (HTCFG_PWM_TM_RELOAD > 65536)
45  #error "HTCFG_PWM_TM_RELOAD out of range! Should be less than or equal to 65536."
46  #endif
```

【引線接法】

- 棕色：GND
- 紅色：VCC 3~7.2V (建議5V)
- 橙色：控制訊號

其工作原理是：

控制信號由接收機的通道進入信號調製晶片，獲得直流偏置電壓。它內部有一個基準電路，產生週期為20ms，寬度為1.5ms的基準信號，將獲得的直流偏置電壓與電位器的電壓比較，獲得電壓差輸出。最後，電壓差的正負輸出到電機驅動晶片決定電機的正反轉。當電機轉速一定時，通過級聯減速齒輪帶動電位器旋轉，使得電壓差為0，電機停止轉動。當然我們可以不用去瞭解它的具體工作原理，知道它的控制原理就夠了。就象我們使用電晶體一樣，知道可以拿它來做開關管或放大管就行了，至於管內的電子具體怎麼流動是可以完全不用去考慮的。

舵機的控制：

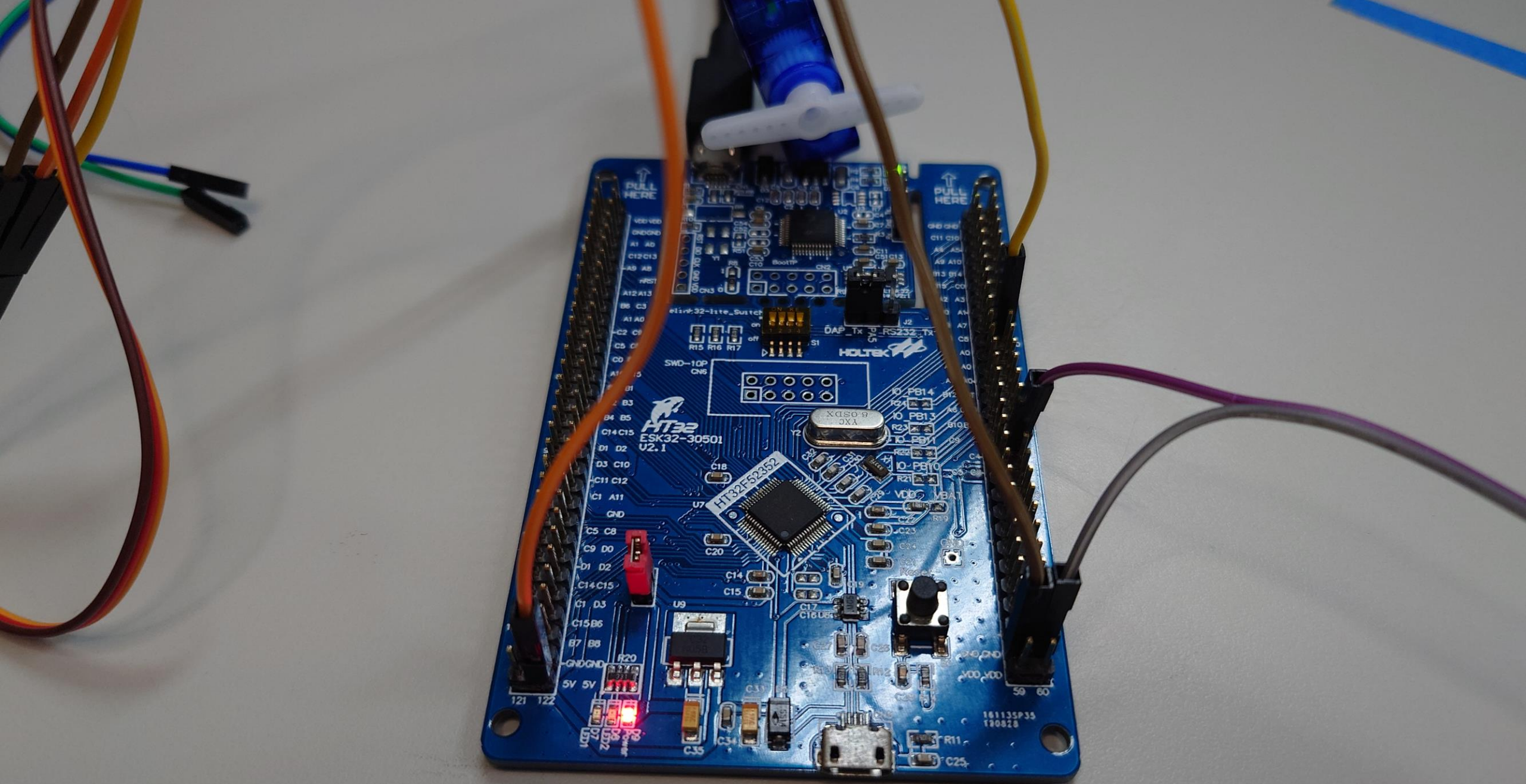
舵機的控制一般需要一個20ms左右的時基脈衝，該脈衝的高電平部分一般為0.5ms~2.5ms範圍內的角度控制脈衝部分。以180度角度伺服為例，那麼對應的控制關係是這樣的：

0.5ms-----	0度；	$0^\circ: 0.5/20=0.025$
1.0ms-----	45度；	$45^\circ: 1.0/20=0.05$
1.5ms-----	90度；	$90^\circ: 1.5/20=0.075$
2.0ms-----	135度；	$135^\circ: 2.0/20=0.1$
2.5ms-----	180度；	$180^\circ: 2.5/20=0.125$

```
#define PWM_DUTY_0      (HTCFG_PWM_TM_RELOAD * 0.025)
#define PWM_DUTY_25     (HTCFG_PWM_TM_RELOAD * )
#define PWM_DUTY_50     (HTCFG_PWM_TM_RELOAD * )
#define PWM_DUTY_75     (HTCFG_PWM_TM_RELOAD * )
#define PWM_DUTY_100    (HTCFG_PWM_TM_RELOAD * )
```

這只是一種參考數值，具體的參數，請參見舵機的技术參數。

$$f = 1/T = 1/20\text{ms} = 50\text{Hz} (= \text{HTCFG_PWM_FREQ_HZ})$$
$$\text{TM_RELOAD} = \text{CRR} = 48000000/50/? < 65535$$



The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. A large red speech bubble is centered on the page, containing the text 'Homework W7-3'.

Homework W7-3

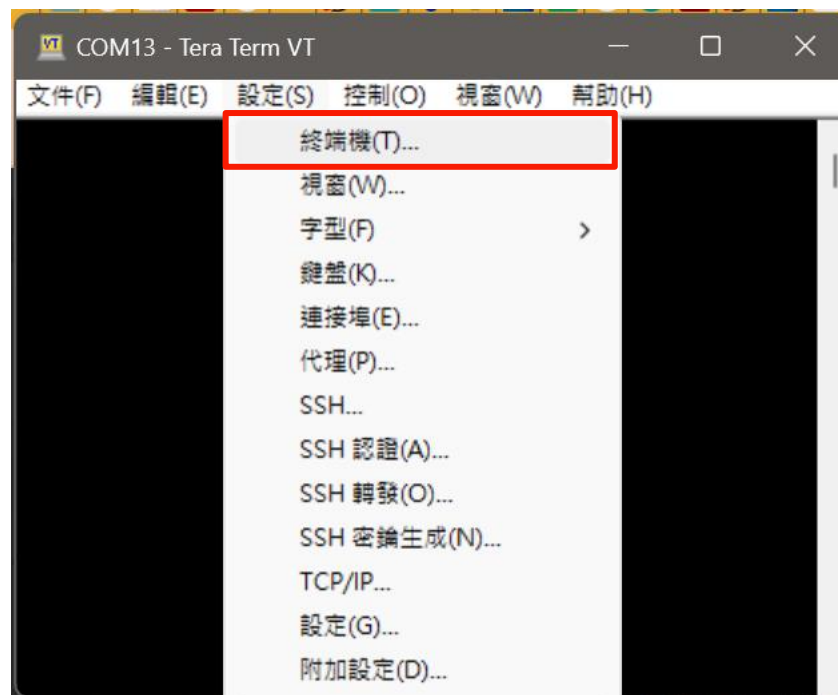
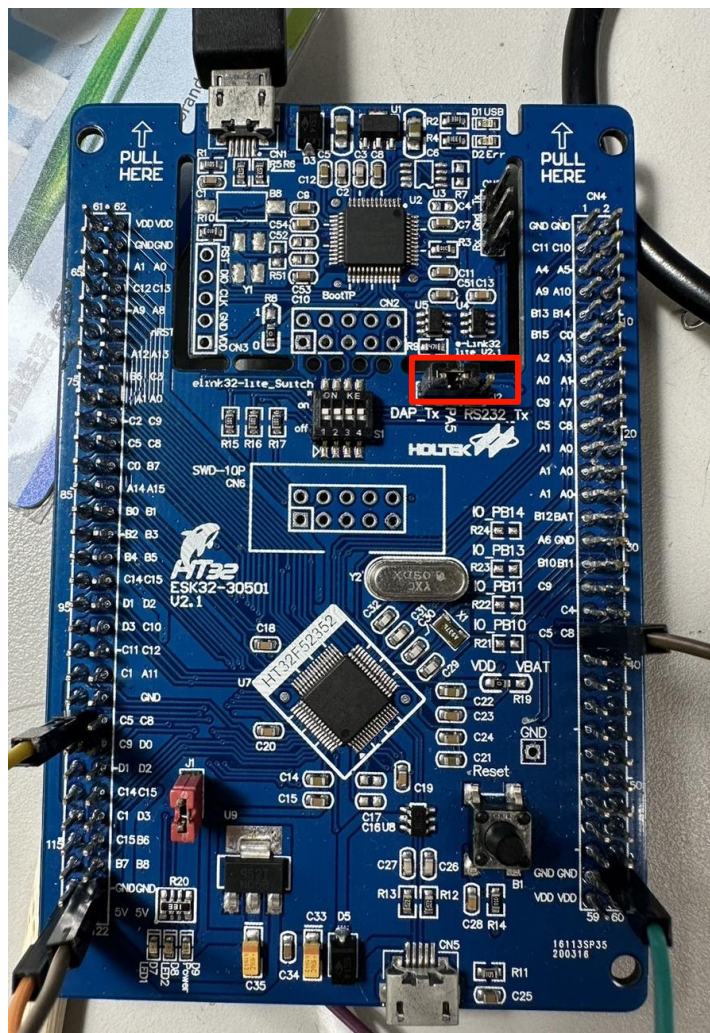
Control the motor angle by input

- Objective: Use laptop keyboard (Tera Term) input Duty Cycle to control SG90.
- Hint:
 1. Connect **CH0** to oscilloscope.
 2. Connect **CH1** to servo motor SG90, and use user input to rotate it to 0, 45, 90, 135, and 180 degrees angle.
 3. Edit code.

☆ PS. Please record.

Step1:

Check if the jumper is connected to DAP_TX and PA5.



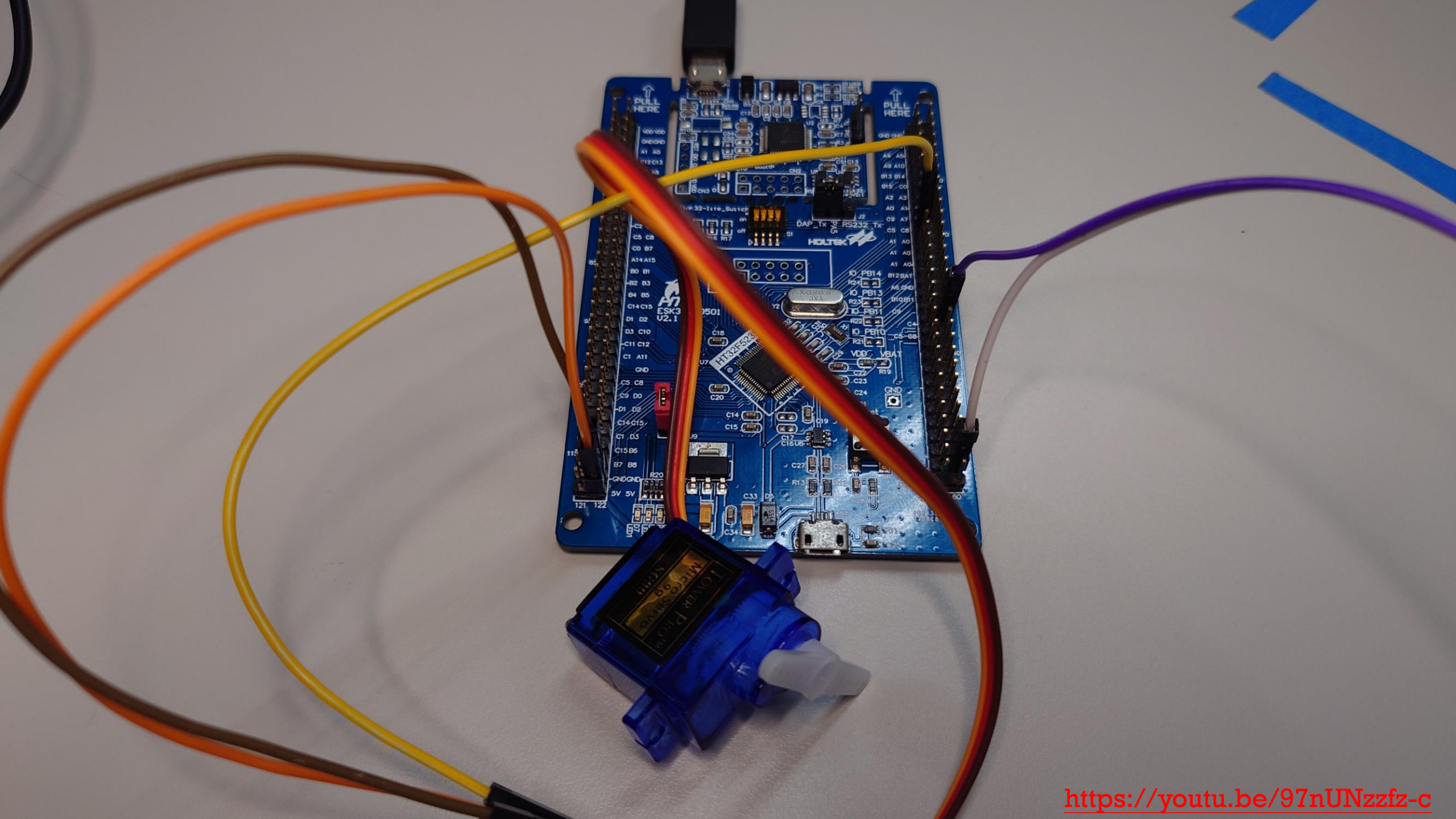
Enable local echo in the terminal settings.

Step2:

- Add code to the appropriate location and annotate or remove unnecessary code.
- PS. input is floating number

```
float input;  
RETARGET_Configuration();  
PWM_Init();
```

```
while (1)  
{  
    printf("Please input ...\n\r");  
    scanf("%f", &input);  
    printf("the input is %f\r\n", input);  
  
    PWM_UpdateDuty(PWM_CH0, HTCFG_PWM_TM_RELOAD * input);  
    PWM_UpdateDuty(PWM_CH1, HTCFG_PWM_TM_RELOAD * input);  
    PWM_Cmd(ENABLE);  
  
    Delay(20000000);  
}
```

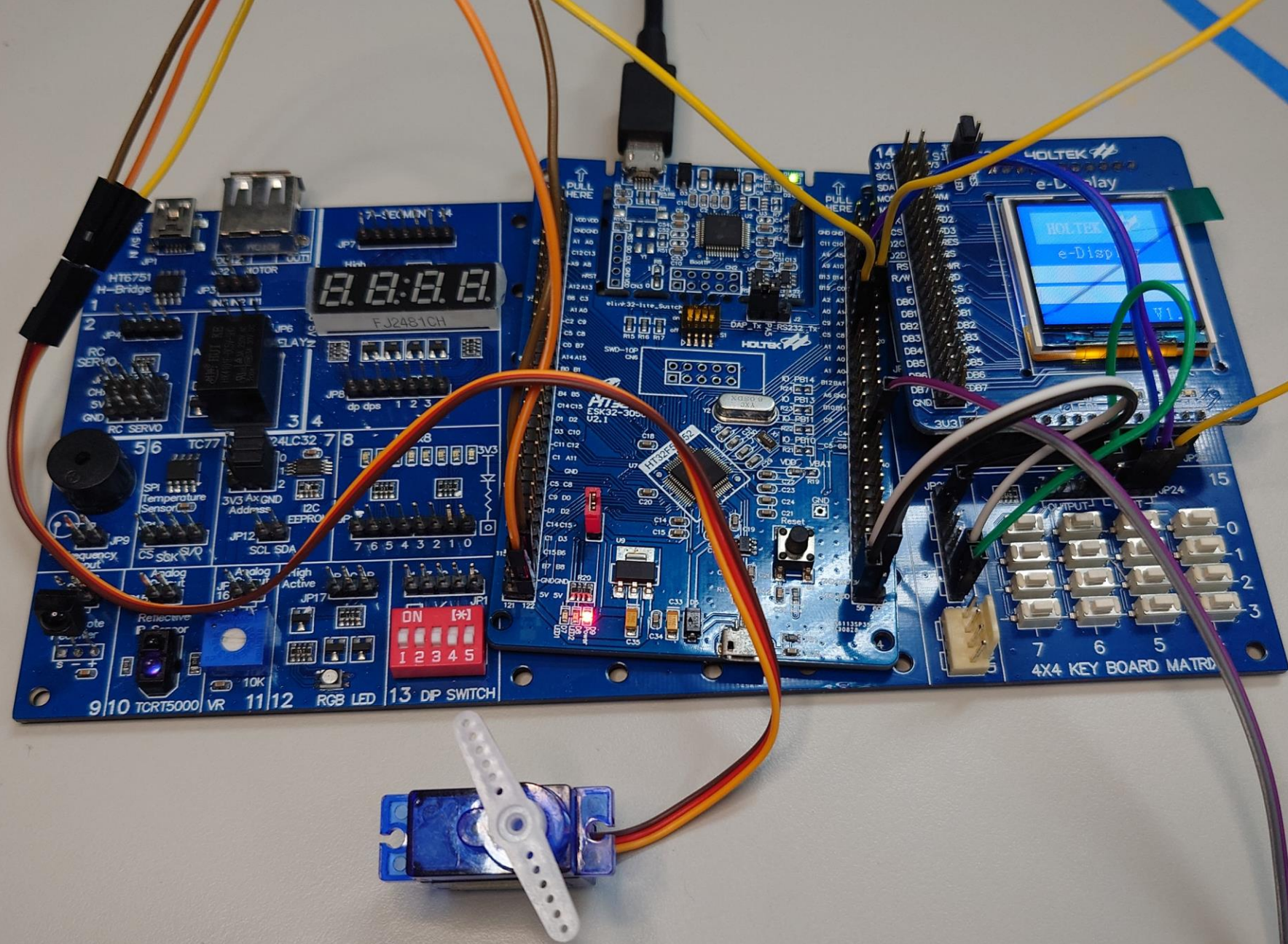
The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. In the center, there is a red speech bubble with a white outline. The text is centered within the bubble.

Homework W7-4 Bonus Question

Bonus: Use buttons to control the motor.

- Objective: Use three buttons to control SG90 to rotate to 0, 90, and 180 degrees.
- Hint:
 1. Connect **CH0** to oscilloscope.
 2. Connect **CH1** to servo motor SG90.
 3. Use pin **A1, A2, A3** to take GPIO input.
 4. Refer to week 2 example GPIO/InputOutput.

☆ PS. Please record and explain the code.





Class
Dismissed