Micro-Controller Experiment

Week10

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. UART introduction
2. UART Project.
3. Homework 10-1.
4. Homework 10-2.
5. Homework 10-3(Bonus).

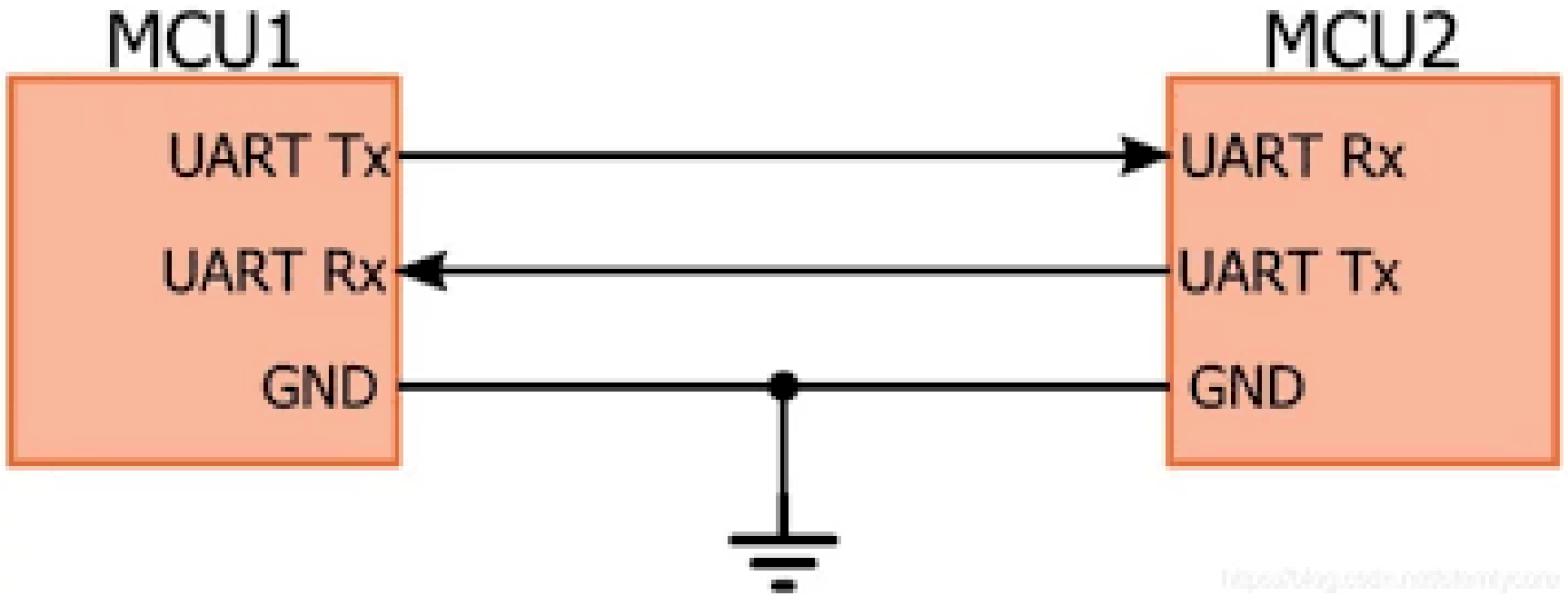
UART Introduction

What is UART?

- Universal Asynchronous Receiver Transmitter (UART) is a computer hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable. Two common signal levels are RS-232.
- UART vs USART:
- UART(Universal Asynchronous Receiver Transmitter): Not Synchronous
- USART(Universal Synchronous Asynchronous Receiver Transmitter): Synchronous

How to transmit?

- URx_RX: Receive data(Input)
- Urx_TX: Send data(Output)



Send signal:

- Start bit: 1bit low potential.
- Word length: 7, 8, or 9-bit
- Stop bit: 1 or 2 bit high potential
- Parity: Even, odd, or no-parity bit generation and detection

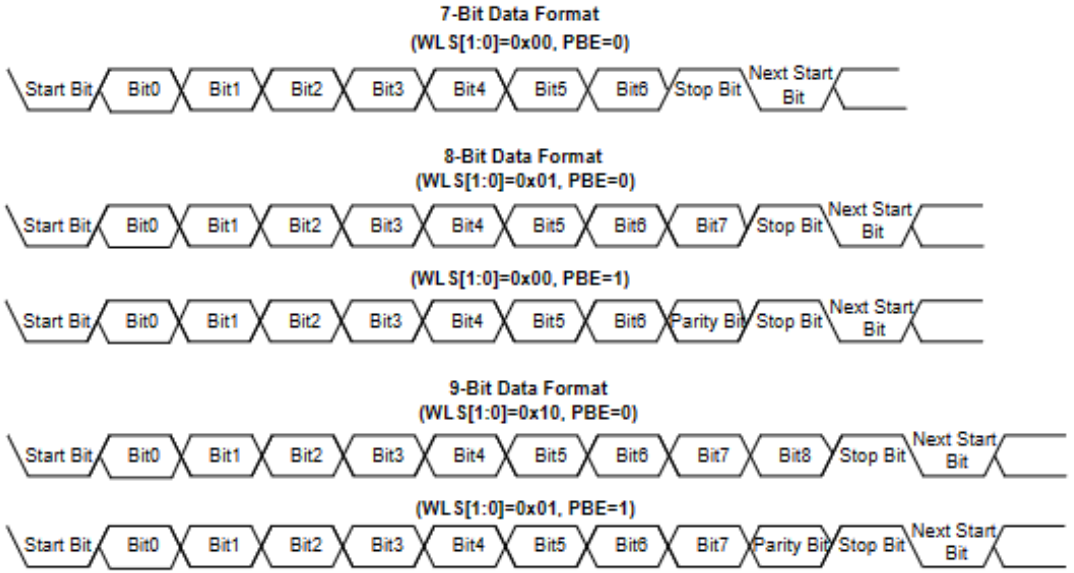
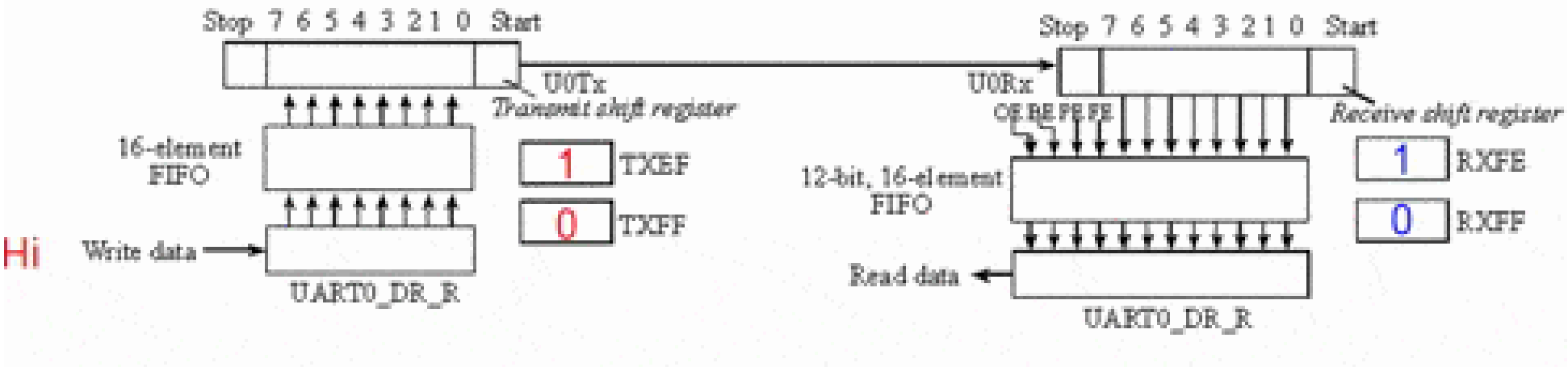


Figure 172. UART Serial Data Format

example



ASCII code: H=DEC 72 => Binary 01001000

i = DEC 105=> Binary 01101001

ASCII code

USART Project

Open project

- Path:“~/HT32_STD_5xxxx_FWLib_V1.5.1_7084/example/USART/Interrupt/MDK_ARMv537”.

int main

```
69  int main(void)
70  {
71    HT32F_DVB_LEDInit(HT_LED1);
72    HT32F_DVB_LEDInit(HT_LED2);
73
74    gURRx_Ptr = gRx_Buffer;
75
76    UxART_Configuration();
77
78    UxART_TxTest();
79
80    while (1)
81    {
82      UxART_RxTest();
83    }
84  }
85
```

UART Configure

UART write test

UART read test

UxART_Configuration

```
90  void UxART_Configuration(void)
91  {
92    #if 0 // Use following function to configure the IP clock speed.
93    // The UxART IP clock speed must be faster 16x then the baudrate.
94    CKCU_SetPeripPrescaler(CKCU_PCLK_UxARTn, CKCU_APBCLKPRE_DIV2);
95    #endif
96
97    { /* Enable peripheral clock of AFIO, UxART                                */
98      CKCU_PeripClockConfig_TypeDef CKCUClock = {{0}};
99      CKCUClock.Bit.AFIO                    = 1;
100     CKCUClock.Bit.HTCFG_UART_RX_GPIO_CLK = 1;
101     CKCUClock.Bit.HTCFG_UART_IPN          = 1;
102     CKCU_PeripClockConfig(CKCUClock, ENABLE);
103   }
104
105   /* Turn on UxART Rx internal pull up resistor to prevent unknow state     */
106   GPIO_PullResistorConfig(HTCFG_UART_RX_GPIO_PORT, HTCFG_UART_RX_GPIO_PIN, GPIO_PR_UP);
107
108   /* Config AFIO mode as UxART function.                                    */
109   AFIO_GPxConfig(HTCFG_UART_TX_GPIO_ID, HTCFG_UART_TX_AFIO_PIN, AFIO_FUN_USART_UART);
110   AFIO_GPxConfig(HTCFG_UART_RX_GPIO_ID, HTCFG_UART_RX_AFIO_PIN, AFIO_FUN_USART_UART);
111
```

CKCU Configure

set pull-up resistor

Pin Configure

UxART_Configuration

```
120     /* !!! NOTICE !!!
121        Notice that the local variable (structure) did not have an initial value.
122        Please confirm that there are no missing members in the parameter settings below in this function.
123     */
124     USART_InitTypeDef USART_InitStructure = {0};
125     USART_InitStructure.USART_BaudRate = 115200;
126     USART_InitStructure.USART_WordLength = USART_WORDLENGTH_8B;
127     USART_InitStructure.USART_StopBits = USART_STOPBITS_1;
128     USART_InitStructure.USART_Parity = USART_PARITY_NO;
129     USART_InitStructure.USART_Mode = USART_MODE_NORMAL;
130     USART_Init(HTCFG_UART_PORT, &USART_InitStructure);
131   }
132
133   /* Enable UxART interrupt of NVIC                                    */
134   NVIC_EnableIRQ(HTCFG_UART_IRQn);
135
136   /* Enable UxART Rx interrupt                                         */
137   USART_IntConfig(HTCFG_UART_PORT, USART_INT_RXDR, ENABLE);
138
139   /* Enable UxART Tx and Rx function                                   */
140   USART_TxCmd(HTCFG_UART_PORT, ENABLE);
141   USART_RxCmd(HTCFG_UART_PORT, ENABLE);
142 }
```

Set up USART transmission format.

Enable Rx interrupt

UxART_TxTest

```
148  void UxART_TxTest(void)
149  {
150    gIsTxFinished = FALSE;
151    gURTx_Ptr = (u8 *)gHelloString;
152    gURTx_Length = sizeof(gHelloString) - 1;
153    USART_IntConfig(HTCFG_UART_PORT, USART_INT_TXDE | USART_INT_TXC, ENABLE);
154
155    while (gURTx_Length != 0);      // Latest byte move to UxART shift register, but the transmission may be on going.
156    while (gIsTxFinished == FALSE); // Set by TXC interrupt, transmission is finished.
157  }
```

Change the Tx transmission completion status to FALSE.

Set up the transmission data and data length.

Enable Tx interrupt

UxART_RxTest

```
163  void UxART_RxTest(void)
164  {
165    u32 i;
166    u32 uLength;
167
168    /* Waiting for receive 5 data
169    if (gURRx_Length >= 5)
170    {
171      // Process Rx data by gRx_Buffer[] and gURRx_Length here
172      // .....
173
174      uLength = gURRx_Length;
175      for (i = 0; i < uLength; i++)
176      {
177        gTx_Buffer[i] = gRx_Buffer[i];
178      }
179
180      #if 1 // Loop back Rx data to Tx for test
181      gIsTxFinished = FALSE;
182      gURTx_Ptr = gTx_Buffer;
183      gURTx_Length = uLength;
184      USART_IntConfig(HTCFG_UART_PORT, USART_INT_TXDE | USART_INT_TXC, ENABLE);
185      #endif
186
187      gURRx_Length = 0;
188    }
189  }
```

When the length of the received data is greater than or equal to 5.

Put the received data into the Tx buffer.

Trigger the Tx interrupt to transmit data.

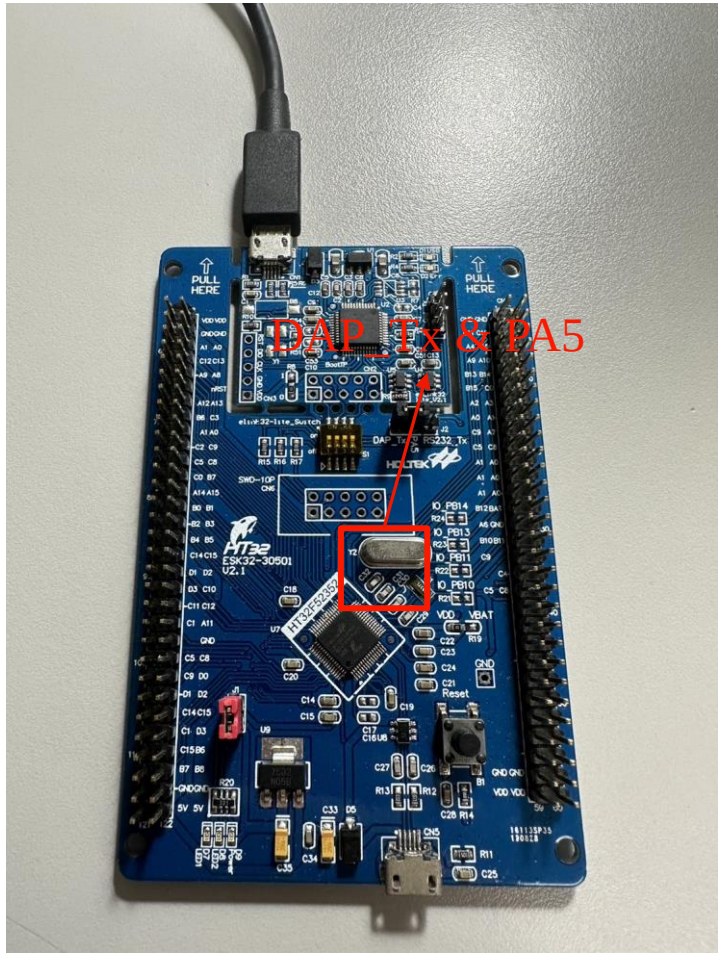
Homework W10-1.

https://github.com/CYCU-AIoT-System-Lab/Microcontroller-Experiment/blob/main/w10/USART-Interrupt-Experiment_Steps.md

Execute the example and explain the code

- Objective: Display data when received exactly or more than 5 characters.
- Hint:

1. Check the jumper and serial port.
2. Do not use RETARGET_Configuration() function.
3. If failed, check Tera Term settings: character encoding, local echo, and baudrate.



☆ PS. Please record.

https://youtu.be/iAOwg1B_CXc

Homework W10-2.

Observe the waveform of the characters.

- Objective: Send two characters and observe the waveform.
- Hint:

1. Wire: Tx -> oscilloscope(RED) / GND -> oscilloscope(BLACK)
2. Edit code:

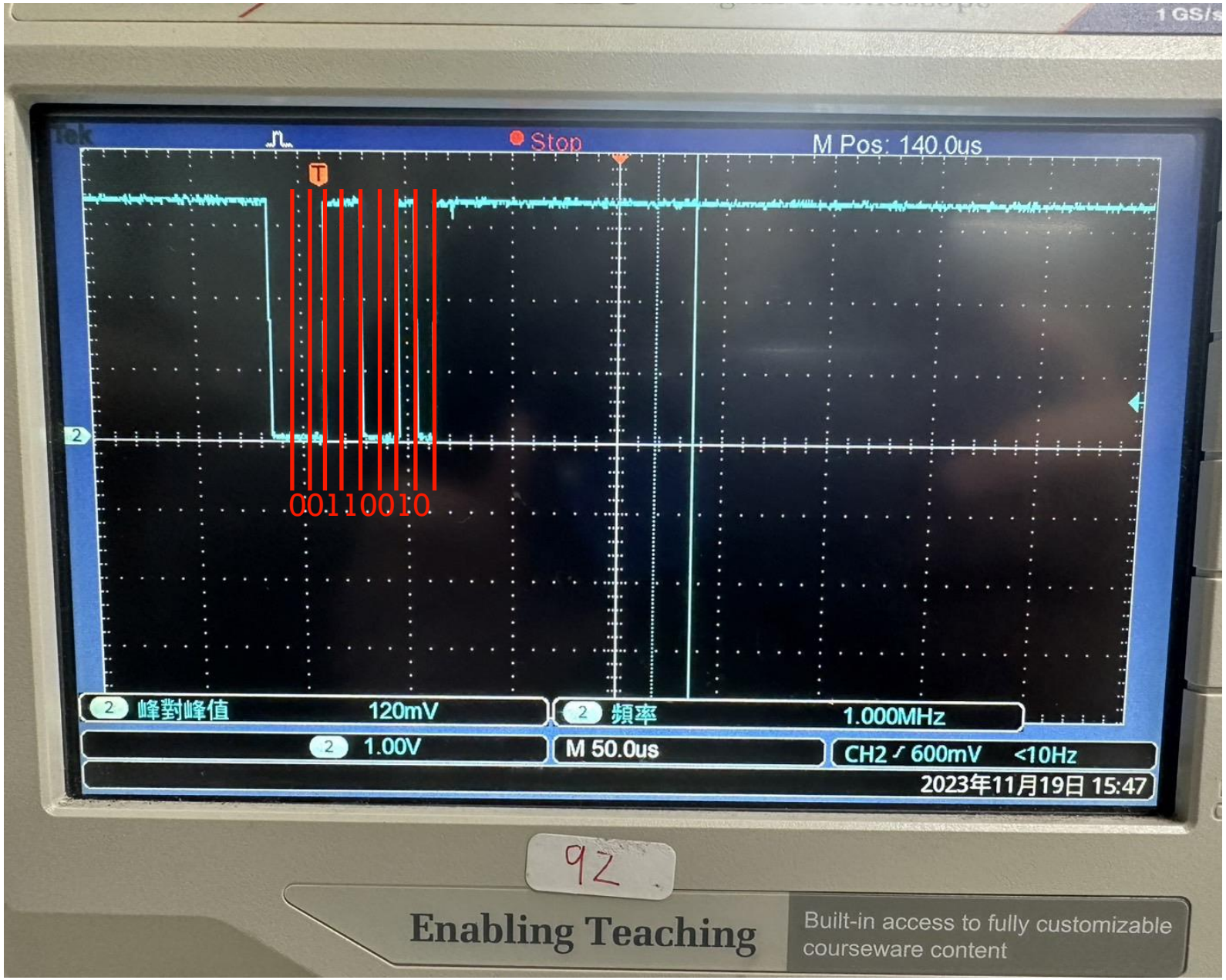
```
void UxART_RxTest(void)
{
  u32 i;
  u32 uLength;

  /* Waiting for receive 5 data
  if (gURRx_Length >= 1)
  {
    // Process Rx data by gRx_Buffer[] and gURRx_Length here
    // .....

    uLength = gURRx_Length;

    for (i = 0; i < uLength; i++)
    {
      gTx_Buffer[i] = gRx_Buffer[i];
    }
```

☆ PS. Please take a picture and explain the result.



二進位、八進位、十進位、十六進位轉換

十進位		二進位	
76	轉換	01001100	轉換
十六進位		八進位	
4c	轉換	114	轉換

ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

theASCIIcode.com.ar

Homework W10-3.

(Bonus)

Entering the student ID, and display your name.

- Objective: Display corresponding name after entering student ID.
- Hint:

1. Don’t use function RETARGET_Configuration() and scanf().
2. Tera Term enable “local echo”.
3. Follow code example on the next page.

☆ PS. Please record.

```
uc8 gHelloString[] = "Hello, this is USART Tx/Rx interrupt example. Please enter your student ID...\r\n";
u8 gTx_Buffer[128];
u8 gRx_Buffer[128];
u8 studentID1[] = "[redacted]";
u8 studentID2[] = "[redacted]";
```

Declaration of member1's student ID array.

Declaration of member2's student ID array.

```
void UxART_RxTest(void)
{
  u32 i;
  u32 uLength;

  /* Waiting for receive 5 data
  if (gURRx_Length >= [redacted])
  {
    // Process Rx data by gRx_Buffer[] and gURRx_Length here
    // .....

    uLength = gURRx_Length;

    for (i = 0; i < uLength; i++)
    {
      gTx_Buffer[i] = gRx_Buffer[i];
      if(gTx_Buffer[i] == studentID1[i])
      {
        if(i==[redacted]){
          printf("\r\n My name is [redacted]\r\n");
        }
      }else if(gTx_Buffer[i] == studentID2[i]){
        if(i==[redacted]){
          printf("\r\n My name is [redacted]\r\n");
       }
      }else{
        printf("\r\nERROR\r\n");
        break;
      }
    }

    #if 1 // Loop back Rx data to Tx for test
    gIsTxFinished = FALSE;
    gURTx_Ptr = (u8 *)gHelloString;
    gURTx_Length = sizeof(gHelloString) - 1;
    USART_IntConfig(HTCFG_UART_PORT, USART_INT_TXDE | USART_INT_TXC, ENABLE);
    #endif

    gURRx_Length = 0;
  }
}
```

length of student numbers

The last element of the array "gTx_Buffer[i]"、" studentID1[i]"

The last element of the array "gTx_Buffer[i]"、" studentID2[i]"

https://youtu.be/sFXiJXmTTYE

Class Dismissed