Holtek – RAM Data Memory

- The **Data Memory is a volatile area** of **8-bit wide RAM internal memory** and is the location where temporary information is stored. The overall Data Memory is subdivided into several banks.
- **Divided into two sections**, the **first** of these is an area of RAM, known as the **Special Function Data Memory**.
  - ⇨Here are located registers which are necessary for correct operation of the device.
  - ⇨**Many** of these registers **can be read from and written to** directly under program control, however, **some remain protected from user manipulation**.
- The **second** area of Data Memory is known as the **General Purpose Data Memory**, which is reserved for general purpose use.
  - ⇨**All locations** within this area are read and write accessible under program control.

Device	Capacity	Banks
HT66F20	64×8	0: 60H~7FH 1: 60H~7FH
HT66F30	96×8	0: 60H~7FH 1: 60H~7FH 2: 60H~7FH
HT66F40	192×8	0: 80H~FFH 1: 80H~BFH
HT66F50	384×8	0: 80H~FFH 1: 80H~FFH 2: 80H~FFH
HT66F60	576×8	0: 80H~ 1: 80H~FFH 2: 80H~FFH 3: 80H~FFH 4: 80H~BFH

	Bank 0, 1, 2		Bank 0, 1, 2		Bank 0, 2	Bank 1		Bank 0, 2	Bank 1
00H	IAR0	20H	PC	40H	Unused	EEC	60H	Unused	
01H	MP0	21H	PCC	41H	EEA		61H	Unused	
02H	IAR1	22H	PDPU	42H	EED		62H	Unused	
03H	MP1	23H	PD	43H	TMPC0		63H	Unused	
04H	BP	24H	PDC	44H	TMPC1		64H	Unused	
05H	ACC	25H	PEPU	45H	PRM0		65H	Unused	
06H	PCL	26H	PE	46H	PRM1		66H	Unused	
07H	TBLP	27H	PEC	47H	PRM2		67H	Unused	
08H	TBLH	28H	PFPU	48H	TM1C0		68H	Unused	
09H	TBHP	29H	PF	49H	TM1C1		69H	Unused	
0AH	STATUS	2AH	PFC	4AH	TM1C2		6AH	Unused	
0BH	SMOD	2BH	Unused	4BH	TM1DL		6BH	Unused	
0CH	LVDC	2CH	Unused	4CH	TM1DH		6CH	Unused	
0DH	INTEG	2DH	Unused	4DH	TM1AL		6DH	Unused	
0EH	WDTC	2EH	ADRL	4EH	TM1AH		6EH	Unused	
0FH	TBC	2FH	ADRH	4FH	TM1BL		6FH	Unused	
10H	INTC0	30H	ADCR0	50H	TM1BH		70H	Unused	
11H	INTC1	31H	ADCR1	51H	TM2C0		71H	Unused	
12H	INTC2	32H	ACERL	52H	TM2C1		72H	Unused	
13H	Unused	33H	Unused	53H	TM2DL		73H	Unused	
14H	MFI0	34H	CP0C	54H	TM2DH		74H	Unused	
15H	MFI1	35H	CP1C	55H	TM2AL		75H	Unused	
16H	MFI2	36H	SIMC0	56H	TM2AH		76H	Unused	
17H	MFI3	37H	SIMC1	57H	TM2RP		77H	Unused	
18H	PAWU	38H	SIMD	58H	TM3C0		78H	Unused	
19H	PAPU	39H	SIMA/SIMC2	59H	TM3C1		79H	Unused	
1AH	PA	3AH	TM0C0	5AH	TM3DL		7AH	Unused	
1BH	PAC	3BH	TM0C1	5BH	TM3DH		7BH	Unused	
1CH	PBPU	3CH	TM0DL	5CH	TM3AL		7CH	Unused	
1DH	PB	3DH	TM0DH	5DH	TM3AH		7DH	Unused	
1EH	PBC	3EH	TM0AL	5EH	SCOMC		7EH	Unused	
1FH	PCPU	3FH	TM0AH	5FH	Unused		7FH	Unused	

Holtek – Indirect Addressing

❒ Indirect Addressing Registers – IAR0, IAR1

- The Indirect Addressing Registers, **IAR0 and IAR1**, **although having their locations in normal RAM register space**, **do not actually physically exist as normal registers**.
- The **method of indirect addressing for RAM data** manipulation **uses these Indirect Addressing Registers and Memory Pointers**, in contrast to **direct memory addressing, where the actual memory address is specified**.
- ***Actions on the IAR0 and IAR1 registers will result in no actual read or write operation to these registers but rather to the memory location specified by their corresponding Memory Pointers, MP0 or MP1***.
- Acting as a pair, ***IAR0 and MP0 can together access data from Bank 0*** while the ***IAR1 and MP1 register pair can access data from any bank***.
- As the Indirect Addressing Registers are not physically implemented, **reading the Indirect Addressing Registers indirectly will return a result of "00H" and writing to the registers indirectly will result in no operation**.

❒ Memory Pointers – MP0, MP1

- Two Memory Pointers, known as MP0 and MP1 are provided.
- These **Memory Pointers** are **physically implemented in the Data Memory** and **can be manipulated in the same way as normal registers providing a convenient way with which to address and track data**.
- When ***any operation to the relevant Indirect Addressing Registers (IAR0 or IAR1) is carried out***, the ***actual address that the microcontroller is directed to, is the address specified by the related Memory Pointer***.
  - ⇨**MP0**, together with Indirect Addressing Register, **IAR0**, are used to access data from **Bank 0**.
  - ⇨**MP1 and IAR1** are used to **access data from all banks according to BP register**.
  - ⇨***Direct Addressing can only be used with Bank 0, all other Banks must be addressed indirectly using MP1 and IAR1***.

❒ Note that for the HT66F20 and HT66F30 devices, bit 7 of the Memory Pointers is not required to address the full memory space. When bit 7 of the Memory Pointers for HT66F20 and HT66F30 devices is read, a value of "1" will be returned. The following example shows how to clear a section of four Data Memory locations already defined as locations adres1 to adres4.

Holtek – EEPROM Data Memory

Device	Capacity	Address
HT66F20	32×8	00H~1FH
HT66F30	64×8	00H~3FH
HT66F40	128×8	00H~7FH
HT66F50/HT66F60	256×8	00H~FFH

- The **EEPROM Data Memory capacity** varies from **32x8** to **256x8** bits, according to the device selected.
- **Read and Write operations** to the EEPROM are carried out in **single byte operations** using **an address and data register in Bank 0** and **a single control register in Bank 1**.
- **Three registers** control the **overall operation of the internal EEPROM Data Memory**.
  - These are the **address register**, **EEA**, the **data register**, **EED** and a single **control register**, **EEC**.
- As both the **EEA and EED registers** are located in **Bank 0**, they **can be directly accessed in the same way as any other Special Function Register**.
- The **EEC register** however, being located in Bank1, **cannot be directly addressed** and **can only be read from or written to indirectly using the MP1 Memory Pointer and Indirect Addressing Register, IAR1**.
  - Because the **EEC** control register is located at address **40H in Bank 1**, the **MP1 Memory Pointer must first be set to the value 40H and the Bank Pointer register, BP, set to the value, 01H, before any operations on the EEC register are executed**.

EEA Name	HT66F60 → HT66F50	HT66F40 →	HT66F30 →	HT66F20 →	Bit			
	7	6	5	4	3	2	1	0
EEA	D7	D6	D5	D4	D3	D2	D1	D0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

Holtek – EEPROM Data Memory / EEA, EEC

EEA

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	×	×	×	×	×	×	×	×

Bit 7~0 Data EEPROM address Data EEPROM address "x" unknown

EEC

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as "0"

Bit 3 WREN: Data EEPROM Write Enable

0: Disable 1: Enable

This is the Data EEPROM Write Enable Bit which must be set high before Data EEPROM write operations are carried out Clearing this bit to zero will inhibit Data EEPROM write operations.

Bit 2 WR: EEPROM Write Control

0: Write cycle has finished **1: Activate a write cycle**

This is the Data EEPROM Write Control Bit and **when set high by the application program will activate a write cycle**. This bit will ***be automatically reset to zero by the hardware after the write cycle has finished.*** Setting this bit high will have no effect if the WREN has not first been set high.

Bit 1 RDEN: Data EEPROM Read Enable

0: Disable 1: Enable

This is the Data EEPROM Read Enable Bit which must be set high before Data EEPROM read operations are carried out. Clearing this bit to zero will inhibit Data EEPROM read operations.

Bit 0 RD: EEPROM Read Control

0: Read cycle has finished **1: Activate a read cycle**

This is the Data EEPROM Read Control Bit and **when set high by the application program will activate a read cycle**. This bit will be ***automatically reset to zero by the hardware after the read cycle has finished***. Setting this bit high will have **no effect if the RDEN has not first been set high**.

Note: The WREN, WR, RDEN and RD can not be set to "1" at the same time in one instruction. The WR and RD can not be set to "1" at the same time.

Holtek – EEPROM Data Memory / Read, Write Procedures

Reading Data from the EEPROM

1. To read data from the EEPROM, **the read enable bit, RDEN**, in the **EEC** register must first be set high to enable the read function.
2. The **EEPROM address** of the data to be read must then be placed in the **EEA** register.
3. If the **RD bit in the EEC register** is now **set high**, **a read cycle will be initiated**.
   - **Setting the RD bit high will not initiate a read operation if the RDEN bit has not been set**.
4. **When the read cycle terminates**, the **RD bit will be automatically cleared to zero**, after which the data can be read from the EED register.
   - The data will remain in the EED register until another read or write operation is executed.
   - The application program can **poll the RD bit to determine when the data is valid for reading**.

Writing Data to the EEPROM

1. To write data to the EEPROM, **the write enable bit, WREN**, in the **EEC** register must first be set high to enable the write function.
2. The **EEPROM address** of the data to be written must then be placed in the **EEA register** and the **data** placed in the **EED** register.
3. If the **Wrbit in the EEC register** is now **set high**, an **internal write cycle will then be initiated**.
   - **Setting the WR bit high will not initiate a write cycle if the WREN bit has not been set.**
4. **When the write cycle terminates**, the **WR bit will be automatically cleared to zero** by the microcontroller, informing the user that the data has been written to the EEPROM.
   - As the **EEPROM write cycle** is controlled **using an internal timer** whose operation is **asynchronous** to microcontroller system clock, *a certain time will elapse before the data will have been written into the EEPROM*.
   - **Detecting when the write cycle has finished** can be implemented **either by polling the WR bit in the EEC register or by using the EEPROM interrupt**.

An **EEPROM Interrupt request will take place** when the **EEPROM Interrupt request flag, DEF, is set**, which **occurs when an EEPROM Write or Read cycle ends**.

Holtek – EEPROM Data Memory / Read, Write Procedures

```
unsigned short EEPROM_RD(unsigned char ee_addr)
{
        _eea=ee_addr;           //user defined address
        _mp1=0x40;//setup memory pointer MP1, MP1 points to EEC register
        _bp=0x01;               //setup Bank Pointer
        //_rden=1;              //set RDEN bit, enable read operations
        //_rd=1;                //start Read Cycle - set RD bit
        //while(_rd)            //check for read cycle end
        _iar1=0x02; //_rden=1;  //set RDEN bit, enable read operations
        _iar1=0x03; //_rd=1;    //start Read Cycle - set RD bit
        while((_iar1&0x01))     //while(_rd) //check for read cycle end
        {   }
        _iar1=0x00;             //disable EEPROM read/write, i.e. _rden=0; _wren=0
        _bp=0;
        return _eed; //return read data
}
void EEPROM_WR(unsigned char ee_addr, unsigned char ee_data)
{
        _eea=ee_addr;           //user defined address
        _eed=ee_data;           //user defined data
        _mp1=0x40;              //setup memory pointer MP1, MP1 points to EEC register
        _bp=0x01;               //setup Bank Pointer
        //_wren=1;              //set WREN bit, enable write operations
        //_wr=1;                //Start Write Cycle - set WR bit
        //while(_wr)            //check for read cycle end
        _iar1=0x08;             //_wren=1;          //set WREN bit, enable write operations
        _iar1=0x0C;                       //_wr=1;          //Start Write Cycle - set WR bit
        while((_iar1&0x04))               //while(_wr) //check for read cycle end
        {   }
        _iar1=0x00;             //disable EEPROM read/write, i.e. _rden=0; _wren=0
        _bp=0;
}
```

EEC

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Holtek – EEPROM Data Memory / A.C. Characteristics

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{LIRC}	System Clock (LIRC)	5V	Ta=25°C	–10%	32	+10%	kHz
		2.2V~5.5V	Ta=–40°C~+85°C	–50%	32	+60%	kHz
f_{TIMER}	Timer Input Pin Frequency	—	—	—	—	1	f_{SYS}
t_{RES}	External Reset Low Pulse Width	—	—	1	—	—	μs
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	t_{SYS}
t_{LVR}	Low Voltage Width to Reset	—	—	120	240	480	μs
t_{LVD}	Low Voltage Width to Interrupt	—	—	20	45	90	μs
t_{LVDS}	LVDO stable time	—	—	15	—	—	μs
t_{BGS}	VBG Turn on Stable Time	—	—	200	—	—	μs
t_{EERD}	EEPROM Read Time	—	—	—	45	90	μs
t_{EEWR}	EEPROM Write Time	—	—	—	2	4	ms
t_{SST}	System Start-up Timer Period (Wake-up from HALT)	—	f_{SYS}=HXT or LXT	—	1024	—	t_{SYS}
			f_{SYS}=ERC or HIRC	—	15~16	—	
			f_{SYS}=LIRC OSC	—	1~2	—	

Note: $t_{SYS}=1/f_{SYS}$

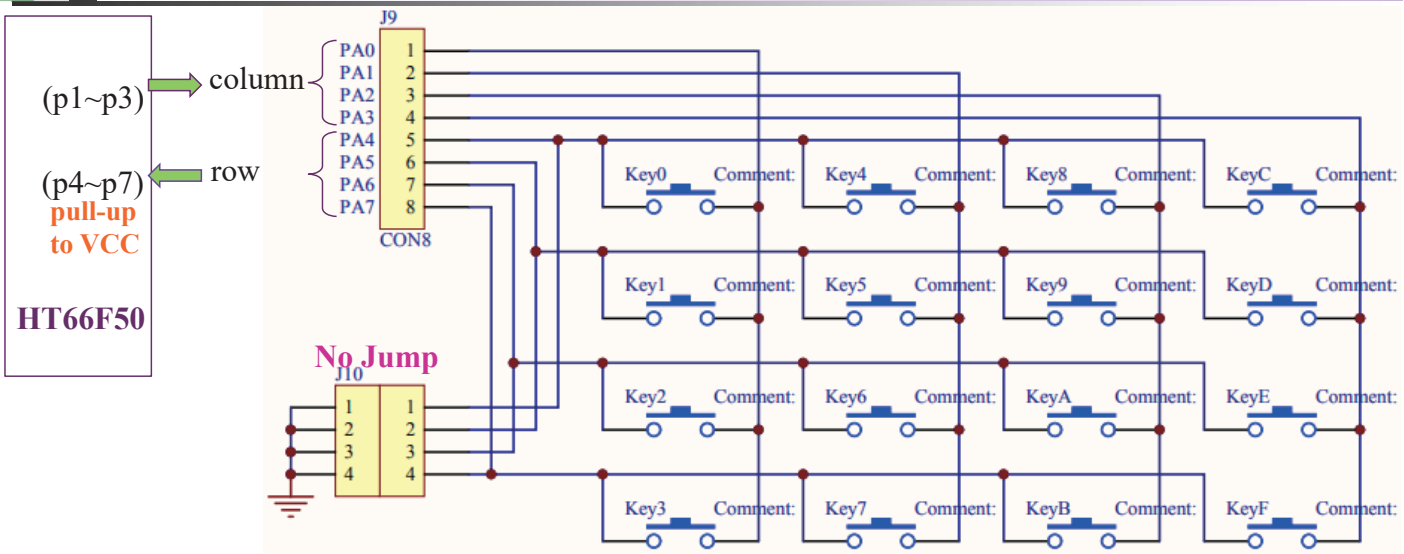
Holtek – RAM Data Memory

- The Data Memory is a volatile area of **8-bit wide RAM internal memory** and is the location where temporary information is stored. The overall Data Memory is subdivided into several banks.
- **Divided into two sections**, the **first** of these is an area of RAM, known as the **Special Function Data Memory**.
  - ⇨Here are located registers which are necessary for correct operation of the device.
  - ⇨**Many** of these registers **can be read from and written to** directly under program control, however, **some remain protected from user manipulation**.
- The **second** area of Data Memory is known as the **General Purpose Data Memory**, which is reserved for general purpose use.
  - ⇨**All locations** within this area are read and write accessible under program control.

Device	Capacity	Banks
HT66F20	64×8	0: 60H~7FH 1: 60H~7FH
HT66F30	96×8	0: 60H~7FH 1: 60H~7FH 2: 60H~7FH
HT66F40	192×8	0: 80H~FFH 1: 80H~BFH
HT66F50	384×8	0: 80H~FFH 1: 80H~FFH 2: 80H~FFH
HT66F60	576×8	0: 80H~ 1: 80H~FFH 2: 80H~FFH 3: 80H~FFH 4: 80H~BFH

	Bank 0, 1, 2		Bank 0, 1, 2		Bank 0, 2	Bank 1		Bank 0, 2	Bank 1
00H	IAR0	20H	PC	40H	Unused	EEC	60H	Unused	
01H	MP0	21H	PCC	41H	EEA		61H	Unused	
02H	IAR1	22H	PDPU	42H	EED		62H	Unused	
03H	MP1	23H	PD	43H	TMPC0		63H	Unused	
04H	BP	24H	PDC	44H	TMPC1		64H	Unused	
05H	ACC	25H	PEPU	45H	PRM0		65H	Unused	
06H	PCL	26H	PE	46H	PRM1		66H	Unused	
07H	TBLP	27H	PEC	47H	PRM2		67H	Unused	
08H	TBLH	28H	PFPU	48H	TM1C0		68H	Unused	
09H	TBHP	29H	PF	49H	TM1C1		69H	Unused	
0AH	STATUS	2AH	PFC	4AH	TM1C2		6AH	Unused	
0BH	SMOD	2BH	Unused	4BH	TM1DL		6BH	Unused	
0CH	LVDC	2CH	Unused	4CH	TM1DH		6CH	Unused	
0DH	INTEG	2DH	Unused	4DH	TM1AL		6DH	Unused	
0EH	WDTC	2EH	ADRL	4EH	TM1AH		6EH	Unused	
0FH	TBC	2FH	ADRH	4FH	TM1BL		6FH	Unused	
10H	INTC0	30H	ADCR0	50H	TM1BH		70H	Unused	
11H	INTC1	31H	ADCR1	51H	TM2C0		71H	Unused	
12H	INTC2	32H	ACERL	52H	TM2C1		72H	Unused	
13H	Unused	33H	Unused	53H	TM2DL		73H	Unused	
14H	MFI0	34H	CP0C	54H	TM2DH		74H	Unused	
15H	MFI1	35H	CP1C	55H	TM2AL		75H	Unused	
16H	MFI2	36H	SIMC0	56H	TM2AH		76H	Unused	
17H	MFI3	37H	SIMC1	57H	TM2RP		77H	Unused	
18H	PAWU	38H	SIMD	58H	TM3C0		78H	Unused	
19H	PAPU	39H	SIMA/SIMC2	59H	TM3C1		79H	Unused	
1AH	PA	3AH	TM0C0	5AH	TM3DL		7AH	Unused	
1BH	PAC	3BH	TM0C1	5BH	TM3DH		7BH	Unused	
1CH	PBPU	3CH	TM0DL	5CH	TM3AL		7CH	Unused	
1DH	PB	3DH	TM0DH	5DH	TM3AH		7DH	Unused	
1EH	PBC	3EH	TM0AL	5EH	SCOMC		7EH	Unused	
1FH	PCPU	3FH	TM0AH	5FH	Unused		7FH	Unused	

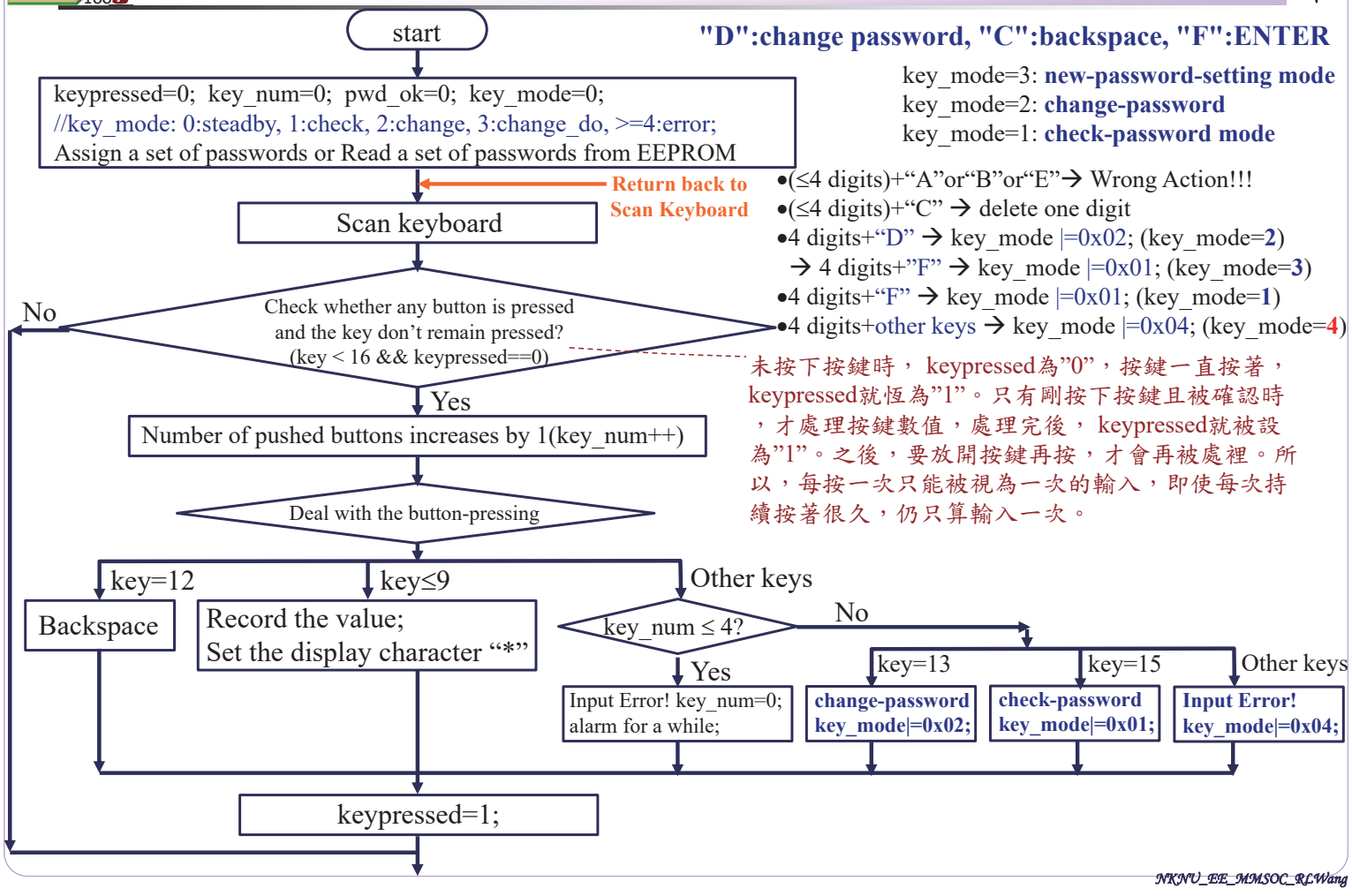
Holtek – Exercise Lab07 / Password Lock (Keyboard Scanning)



Check every row in turn for every "P3P2P1P0" pattern. Increment the value of "key" by 1 after checking every row.
The detection of "0" value indicates that a button is pressed and then the checking procedure is stopped.

	P3P2P1P0			
	1st col	2nd col	3rd col	4th col
	1110	1101	1011	0111
!(_pa & (1<<4)); or !(_pa & 0b00010000); **Check 1st row**	0(0000)	4(0100)	8(1000)	C(1100)
!(_pa & (1<<5)); or !(_pa & 0b00100000); **Check 2nd row**	1(0001)	5(0101)	9(1001)	D(1101)
!(_pa & (1<<7)); or !(_pa & 0b01000000); **Check 3rd row**	2(0010)	6(0110)	A(1010)	E(1110)
!(_pa & (1<<8)); or !(_pa & 0b10000000); **Check 4th row**	3(0011)	7(0111)	B(1011)	F(1111)

Holtek – Exercise Lab07 / Password Lock (Flow Chart)



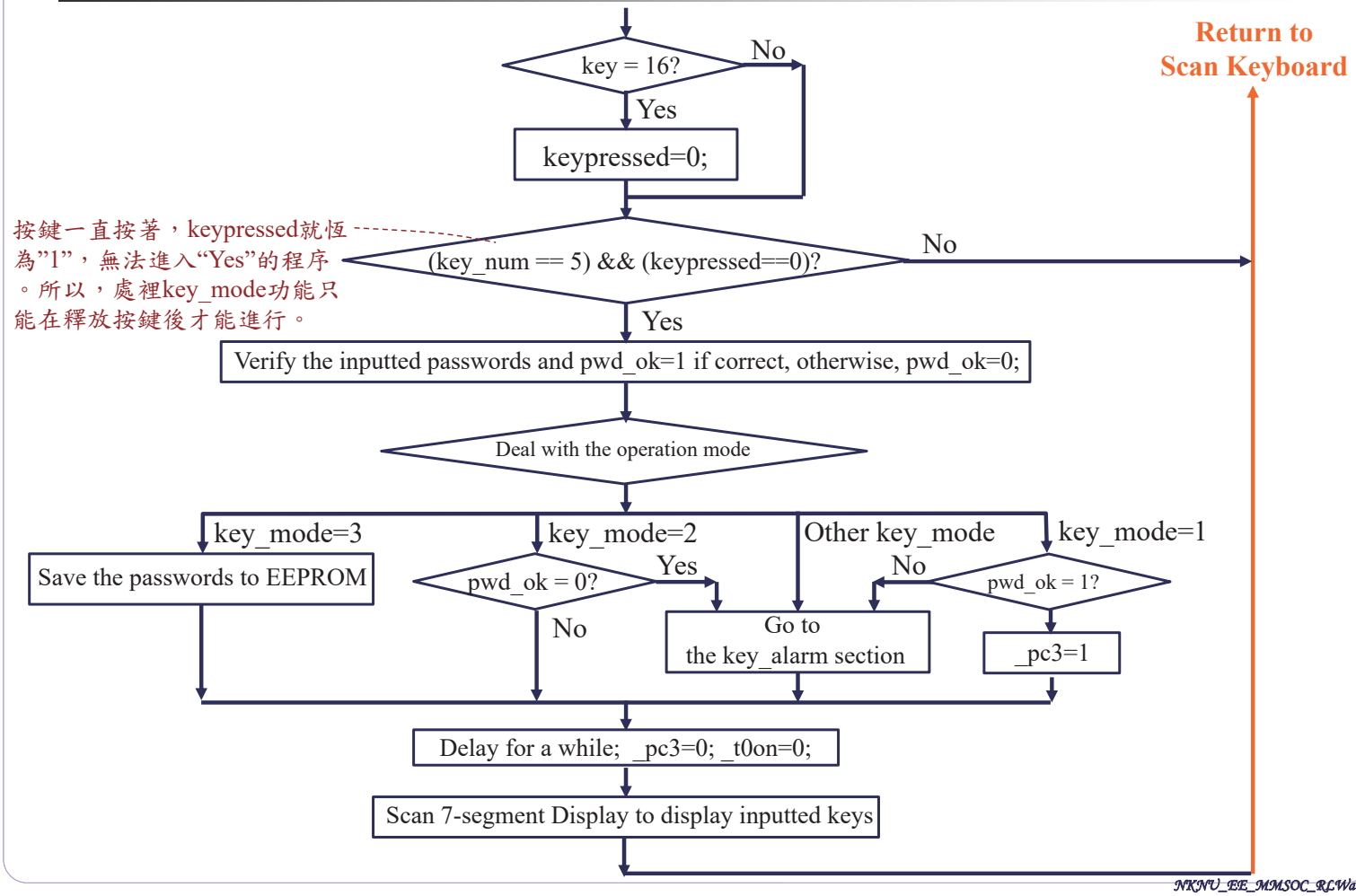
"D":change password, "C":backspace, "F":ENTER

key_mode=3: **new-password-setting mode**
key_mode=2: **change-password**
key_mode=1: **check-password mode**

- (≤4 digits)+"A"or"B"or"E"→ Wrong Action!!!
- (≤4 digits)+"C" → delete one digit
- 4 digits+"D" → key_mode |=0x02; (key_mode=**2**)
  → 4 digits+"F" → key_mode |=0x01; (key_mode=**3**)
- 4 digits+"F" → key_mode |=0x01; (key_mode=**1**)
- 4 digits+other keys → key_mode |=0x04; (key_mode=**4**)

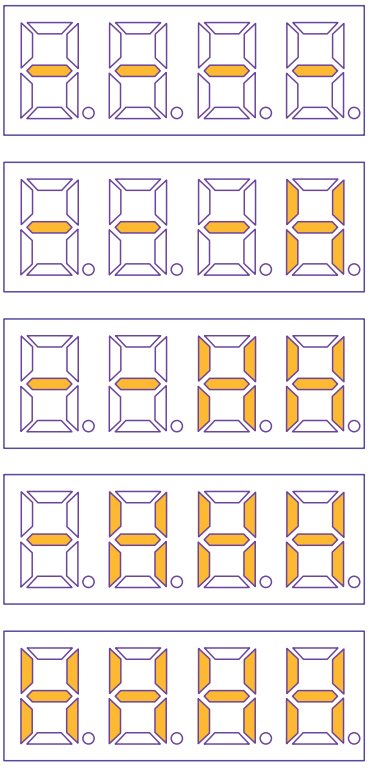
未按下按鍵時，keypressed為"0"，按鍵一直按著，keypressed就恆為"1"。只有剛按下按鍵且被確認時，才處理按鍵數值，處理完後，keypressed就被設為"1"。之後，要放開按鍵再按，才會再被處裡。所以，每按一次只能被視為一次的輸入，即使每次持續按著很久，仍只算輸入一次。

Holtek – Exercise Lab07 / Password Lock (Flow Chart)



Holtek – Exercise Lab07 / Password Lock (Flow Chart)

正常輸入



輸入新密碼