PIC12F629 / 675
Programming PIC in C
Part II

RNDr. Vojtěch Krmíček
vojtec@ics.muni.cz

Ing. Zbyněk Bureš, Ph.D.
zbynek.bures@unob.cz

Inline Assembler

- Two variants:

```
unsigned int var;
#asm
bcf 0,3
rlf _var
rlf _var+1
#endasm
```

```
// or like this
asm("bcf 0,3");
asm("rlf _var");
asm("rlf _var+1");
```

- A #asm block can’t be used within any C constructs such as if, while, do...

Timing Functions

Frequency of oscillator must be defined:

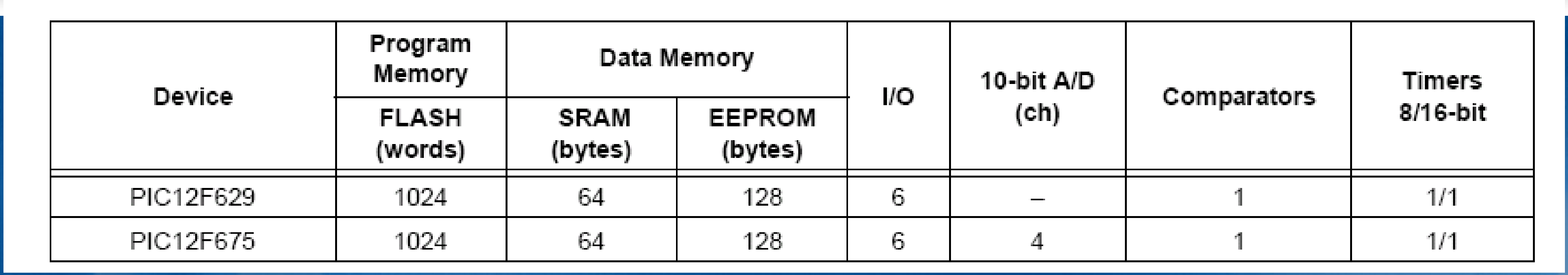
```
#define _XTAL_FREQ 4000000
```

Two functions:

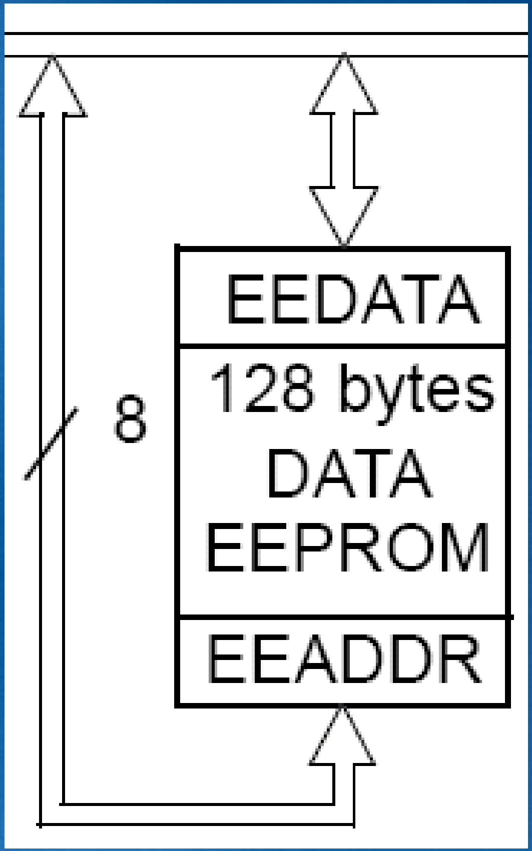
❿ __delay_ms(x) // request a delay in milliseconds

❿ __delay_us(x) // request a delay in microseconds

Package



Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	FLASH (words)	SRAM (bytes)	EEPROM (bytes)				
PIC12F629	1024	64	128	6	–	1	1/1
PIC12F675	1024	64	128	6	4	1	1/1



EEPROM Programming

- Readable and writable during normal voltage,
- memory is not directly accessible, it’s mapped in the register file space,
- direct access via C functions:
  - eeprom_write()
  - eeprom_read()
- address range from 0h to 7Fh
- interrupt on write complete (EEIF)

Functions for Accessing EEPROM I

- eeprom_write() initiates process of writing to the EEPROM memory and returns when write is completed
- new data in EEPROM are valid approx. 4ms later (= 4000 instruction cycles!)
- but next read/write operation waits until previous one is finished

Functions for Accessing EEPROM II

```
#include <htc.h>
void eetest(void){
  unsigned char value = 1;
  unsigned char address = 0;
  eeprom_write(address,value);
// Initiate writing value to address
  value = eeprom_read(address);
// read from EEPROM at address
}
```