
 Kompatibiln&#65533;zapojen&#65533; LCD panel s ATmega8
 Ke sta&#65533;en&#65533; wait.c == wait.pdf == wait.htm
 .
 .
 <hr /> Untitled <!-- body { color: #000000; background-color: #FFFFFF; } .cpp1-assembler { } .cpp1-brackets { } .cpp1-comment { color: #008000; font-style: italic; } .cpp1-float { color: #000080; } .cpp1-hexadecimal { color: #000080; } .cpp1-character { } .cpp1-identifier { } .cpp1-illegalchar { } .cpp1-number { color: #000080; } .cpp1-octal { color: #0000FF; } .cpp1-preprocessor { } .cpp1-reservedword { font-weight: bold; } .cpp1-space { color: #008080; } .cpp1-string { color: #800000; } .cpp1-symbol { } --> /*---------- soubor: wait.c verze: 1.0 datum: 9.1.2012 popis: Cekaci rutiny a makra pro bitove operace Cekaci rutiny jsou mnohem uspornejsi nez bezne "delay.h" Tento soubor vynikl ze souboru "mojelib.h" jehoz autora bohuzel neznam. ----------*/ // Zmena bitu portu, IO registru nebo promenne: #define setb(bajt,bit) bajt |= 1<<(bit) //nastav bit #define clrb(bajt,bit) bajt &= ~(1<<(bit)) //nuluj bit #define negb(bajt,bit) bajt ^= 1<<(bit) //neguj bit // Casove smycky: void wait_ms (unsigned int c); // cekej milisekund(max65553): void wait_us (unsigned int c); // cekej mikrosekund(max65553): //cekej milisekund: <span

class="cpp1-reservedword">unsigned <span
class="cpp1-reservedword">char <span
class="cpp1-identifier">reg21 <span
class="cpp1-symbol">= <span
class="cpp1-identifier">F_CPU <span
class="cpp1-symbol">/ <span
class="cpp1-number">60000;<span
class="cpp1-space"> // F_CPU = frekv. oscilatoru [Hz],
  <span
class="cpp1-comment">// (vnitrni konstanta prekladace) <span
class="cpp1-reservedword">void <span
class="cpp1-identifier">wait_ms <span
class="cpp1-brackets">(unsigned<span
class="cpp1-space"> int<span
class="cpp1-space"> c) { <span
class="cpp1-space"> asm<span
class="cpp1-assembler">("push r20");   <span
class="cpp1-reservedword">asm("push r21");
asm<span
class="cpp1-assembler">("_Wms0:");   <span
class="cpp1-reservedword">asm("ldi r20,0x14");
asm<span
class="cpp1-assembler">("_Wms1:");   <span
class="cpp1-reservedword">asm("lds r21,reg21");
asm<span
class="cpp1-assembler">("_Wms2:");   <span
class="cpp1-reservedword">asm("dec r21");
 <span
class="cpp1-reservedword">asm("brne _Wms2");
 <span
class="cpp1-reservedword">asm("dec r20");
 <span
class="cpp1-reservedword">asm("brne _Wms1");
 <span
class="cpp1-reservedword">asm("dec r24");
 <span
class="cpp1-reservedword">asm("brne _Wms0");
 <span
class="cpp1-reservedword">asm("dec r25");
 <span
class="cpp1-reservedword">asm("brpl _Wms0");
 <span
class="cpp1-reservedword">asm("pop   r21");
 <span
class="cpp1-reservedword">asm("pop   r20"); }
//cekej mikrosekund: <span

class="cpp1-reservedword">unsigned char reg20 = F_CPU / 6000000 + 1; void wait_us(unsigned int c) {   asm("push r20");asm("_wus0:");   asm("lds r20,reg20"); //1-6MHz: r20=1 6-12MHz: r20=2 asm("_wus1:"); asm("dec r20"); asm("brne _wus1"); asm("dec r24"); asm("brne _wus0"); asm("dec r25"); asm("brpl _wus0"); asm("pop r20"); }//eof //(c) OK1ZKV 2012