New Jersey Institute of Technology

Special Topics for Embedded Systems

(Balancing memory usage, speed, performance)

Saving Memory Space

- RAM and ROM are not interchangeable – design your software accordingly

- RAM
  - Recall memory allocations for chars and ints – 2x the RAM needed!
  - Compiler/linker attempts to conserve RAM, cannot for structs
  - Some compilers don’t allocate if not used

Optimizing ROM

ROM Need to Initialize Variables

```
struct sMyStruct {
   unsigned char iMember;
};
struct sMyStruct a_sMyData[3];
struct sMyStruct *p_sMyData;
int   i;

void main( void )
{
   a_sMyData[0].iMember = 0;              /* Method 1 */
   a_sMyData[1].iMember = 5;
   a_sMyData[2].iMember = 10;

   for (i=0; i<3; ++i)                    /* Method 2 */
      a_sMyData[0].iMember = 5 * i;

   i=0;                                   /* Method 3 */
   p_sMyData = a_sMyData;
   do
   {
      p_sMyData->iMember = i;
      i += 5;
      ++p_sMyData;
   }
   while (i<10);
}
```

Initialization Method 1

ROM: 0x16 Bytes

```
a_sMyData[0].iMember = 0;
a_sMyData[1].iMember = 5;
a_sMyData[2].iMember = 10;
```

```
00F81C   43C2 0204          clr.b   &a_sMyData
  a_sMyData[1].iMember = 5;
00F820   40F2 0005 0205     mov.b   #0x5,&0x205
  a_sMyData[2].iMember = 10;
00F826   40F2 000A 0206     mov.b   #0xA,&0x206
  for (i=0; i<3; ++i)
00F82C   4382 0202          clr.w   &i
```

Initialization Method 2

ROM: 0x22 Bytes

```
for (i=0; i<3; ++i)
   a_sMyData[0].iMember = 5 * i;
```

```
  for (i=0; i<3; ++i)
00F82C   4382 0202          clr.w   &i
00F830   3C0A               jmp     0xF846
    a_sMyData[0].iMember = 5 * i;
00F832   425E 0202          mov.b   &i,R14
00F836   4E4F               mov.b   R14,R15
00F838   5E4E               rla.b   R14
00F83A   5E4E               rla.b   R14
00F83C   5F4E               add.b   R15,R14
00F83E   4EC2 0204          mov.b   R14,&a_sMyData
  for (i=0; i<3; ++i)
00F842   5392 0202          inc.w   &i
  for (i=0; i<3; ++i)
00F846   90B2 0003 0202     cmp.w   #0x3,&i
00F84C   3BF2               jl      0xF832
  i=0;
00F84E   4382 0202          clr.w   &i
```

Initialization Method 3 ROM: 0x26 Bytes

```
i=0;
p_sMyData = a_sMyData;
do
{
    p_sMyData->iMember = i;
    i += 5;
    ++p_sMyData;
}
while (i<10);
```

```
  i=0;
00F84E   4382 0202          clr.w   &i
  p_sMyData = a_sMyData;
00F852   40B2 0204 0200     mov.w   #0x204,&p_sMyData
     p_sMyData->iMember = i;
00F858   421F 0200          mov.w   &p_sMyData,R15
00F85C   42DF 0202 0000     mov.b   &i,0x0(R15)
     i += 5;
00F862   50B2 0005 0202     add.w   #0x5,&i
     ++p_sMyData;
00F868   5392 0200          inc.w   &p_sMyData
  while (i<10);
00F86C   90B2 000A 0202     cmp.w   #0xA,&i
00F872   3BF2               jl      0xF858
}
00F874   4130               ret
```

Balancing ROM and Speed

- Consider three different snippets of code that store values to an array of 10 elements

Fastest? Smallest? Largest?

```
#define  ARRAY_SIZE  10

unsigned int array[ARRAY_SIZE];
unsigned int array2[ARRAY_SIZE];
```

```
void sample1(void)
{
  char  i;

  for (i=0; i<ARRAY_SIZE; i++)
  {
     array[i] = (array2[i] * 5)/40;
  }
}
```

```
void sample2(void)
{
  char  i;

  for (i=0; i<ARRAY_SIZE; i++)
  {
     array[i] = array2[i]/8;
  }
}
```

```
void sample3(void)
{
  array[0] = array2[0]/8;
  array[1] = array2[1]/8;
  array[2] = array2[2]/8;
  array[3] = array2[3]/8;
  array[4] = array2[4]/8;
  array[5] = array2[5]/8;
  array[6] = array2[6]/8;
  array[7] = array2[7]/8;
  array[8] = array2[8]/8;
  array[9] = array2[9]/8;
}
```

IAR Workbench - Setup

- batterylife.c
- Use the simulator, 2013.
- Must cause linker to generate output information
  - Project->Options->C/C+ Compiler…Output tab…check Generate Debug Information
  - Project->Options->C/C+ Compiler…List tab…select Output List File
  - Project->Options->Assembler …Output tab…check Generate Debug Information

Checking Mem and Performance

- After compiling, open the MAP file for memory size allocations
- Start debugger, then launch profiler
  - View->Profiling
  - Must enable the profiler (Left-most “power” button)
  - Run program and then check values

	ROM Size (bytes)	Execution time (cycles)
sample1()		
sample2()		
sample3()		