A close look at our simple initial C program as it exists in the HCS08QG8 microcontroller (details may differ for the SH8 microcontroller.):

C code (comments stripped out):

```
void main(void){
   short i=0;
   PTBDD_PTBDD6=1;
   for(;;){
      __RESET_WATCHDOG();
      PTBD_PTBD6=0;
      i++;
     PTBD_PTBD6=1;
   }
}
```

Disassembled equivalent:

```
E092  AIS   #-2          ;Add immediate value (-2) to stack pointer (to make space for i
E094  TSX                ;Transfer the value in the stack ponter (+1?) to HX
E095  CLR  1,X           ;Clear the byte in RAM pointed to by X+1 (least sign byte of i)
E097  CLR  ,X            ;Clear the byte in RAM pointed to by X (most significant byte of i)
E098  BSET  6,0x03   ;Bit Set at address 0x03 (PTABDD) bit 6 – makes it an output
E09A  STA  0x1800    ;Store whatever is in "A" (it's zero) to address 1800 (feeds dog)
E09D  BCLR  6,0x02 ;Bit Clear at address 0x02 (PTABD) bit 6
E09F  INC  1,X           ;Increment byte pointed to by X+1 (least significant byte of i)
E0A1  BNE  *+3    ; abs= 0xE0A4  Branch to E0A4 if prev. operation didn't give zero.
E0A3  INC ,X             ;Increment byte pointed to by X (most significant byte of i)
E0A4  BSET  6,0x02  ;Bit Set at 0x02 (PTBD) bit 6
E0A6  BRA    *-12    ;abs=0xE09A  Branch always to address E09A
E0A8  BRSET  0,0x00 *-83   ;abs=0xE055  Branch if bit 0 at address 0 is set to E055.
```

Machine language (hexadecimal) in memory:

```
E090    00 00 A7 FE 95 6F 01 7F      AIS   #-2;    TSX;    CLR  1,X;    CLR  ,X
E098    1C 03 C7 18 00 1D 02 6C      BSET  6,0x03; STA  0x1800; BCLR  6,0x02; INC
E0A0    01 26 01 7C 1C 02 20 F2      1,X; BNE  *+3; INC ,X; BSET  6,0x02; BRA *-12
E0A8    00 00 uu uu uu uu uu uu      BRSET  0,0x00 *-83  (we should never get to this)
```

In memory:

Data:1 (globals)

```
_PTBDD   address 0x3
_SRS       address 0x1800
_PTBD      address 0x2
```

Data:2 (locals – on stack)

```
i       address 0x14E
```

A close look at our simple initial C program, with I put in as a global instead of a local variable, +- as it exists in the microcontroller:

C code (comments stripped out):

```
short i=0;
void main(void){
PTBDD_PTBDD6=1;
   for(;;){
      __RESET_WATCHDOG();
      PTBD_PTBD6=0;
      i++;
     PTBD_PTBD6=1;
   }
}
```

Disassembled equivalent:

```
E092  BSET  6,0x03   ;Bit Set at address 0x03 (PTABDD) bit 6 – makes it an output
E094  STA  0x1800    ;Store whatever is in "A" (it's zero) to address 1800 (feeds dog)
E097  BCLR  6,0x02  ;Bit Clear at address 0x02 (PTABD) bit 6
E099  LDHX  #0x0100 ;Load HX with pointer to i (address 0x100)
E09C  INC 1,X          ;Increment byte pointed to by X+1 (least significant byte of i)
E09E  BNE  *+3       ; abs= 0xE0A1  Branch to E0A1 if prev. operation didn't give zero.
E0A0  INC ,X           ;Increment byte pointed to by X (most significant byte of i)
E0A1  BSET  6,0x02  ;Bit Set at 0x02 (PTBD) bit 6
E0A3  BRA    *-15    ;abs=0xE094  Branch always to address E094
E0A8  BRSET  0,0x00 *-86   ;abs=0xE04F  Branch if bit 0 at address 0 is set to E04F.
```

Machine language (hexadecimal) in memory:

```
E090    00 00 1C 03 C7 18 00 1D      BSET  6,0x03; STA  0x1800; BCLR
E098    02 45 01 00 6C 01 26 01      6,0x02; LDHX #0100;   INC1,X; BNE  *+3
E0A0    01 26 01 7C 1C 02 20 F2      INC ,X; BSET  6,0x02; BRA *-12
E0A8    00 00 uu uu uu uu uu uu      BRSET  0,0x00 *-83  (we should never get to this)
```

In memory:

```
Data:1 (globals)
i       address 0x100
_PTBDD   address 0x3
_SRS        address 0x1800
_PTBD      address 0x2
```

Data:2 (locals – on stack)
None

Code for simple program to light LED to a particular level
In this case, the global variable sets it to “off”.

main.c:

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations
*/

short brightness=0; /*brightness, can vary from 0 to 9999*/

void main(void) {
  short j;
  /*EnableInterrupts;*/ /* enable interrupts */
  /* include your code here */
  PTBDD_PTBDD6=1;
  for(;;) {
    __RESET_WATCHDOG(); /* feeds the dog */
      PTBD_PTBD6=0;               /*turns the LED on*/
      for(j=0;j<brightness;j++){}
      PTBD_PTBD6=1;               /*turns the LED off*/
      for(j=brightness;j<10000;j++){}
  } /* loop forever */
  /* please make sure that you never leave main */
}
```