**Listing 1. Plik nagłówkowy dla asemblera TASM zawierający definicje związane ze sterownikiem MINI\_IO**

; SDC\_One MiniIO ports and field definitions

; JK & gbm 2016-18

#ifndef MINIO\_BASE

#define MINIIO\_BASE 0

#endif

;=====================================

; MINI IO registers

IO\_LEDS **=** MINIIO\_BASE **+** 0

IO\_CON **=** MINIIO\_BASE **+** 1

IO\_TMR **=** MINIIO\_BASE **+** 2

IO\_CSR **=** MINIIO\_BASE **+** 3

IO\_LEDR **=** MINIIO\_BASE **+** 4

IO\_LEDG **=** MINIIO\_BASE **+** 5

IO\_LEDB **=** MINIIO\_BASE **+** 6

IO\_LEDN **=** MINIIO\_BASE **+** 7

IO\_MS\_SEC **=** MINIIO\_BASE **+** 8

IO\_MS\_TRK **=** MINIIO\_BASE **+** 9

IO\_MS\_UNIT **=** MINIIO\_BASE **+** 0ah

IO\_MS\_CMD **=** MINIIO\_BASE **+** 0bh

IO\_MS\_A0 **=** MINIIO\_BASE **+** 0ch

IO\_MS\_A1 **=** MINIIO\_BASE **+** 0dh

IO\_MS\_A2 **=** MINIIO\_BASE **+** 0eh

;=====================================

; LED control

#define IO\_LEDS\_RED **(**1 **<<** 0**)**

#define IO\_LEDS\_GRN **(**1 **<<** 1**)**

#define IO\_LEDS\_BLU **(**1 **<<** 2**)**

#define IO\_LEDS\_NUC **(**1 **<<** 3**)**

#define IO\_LEDS\_SET **(**1 **<<** 6**)**

#define IO\_LEDS\_RES **(**1 **<<** 7**)**

#define IO\_LEDS\_TGL **(**3 **<<** 6**)**

; timer period 8ms flag

#define IO\_TMR\_X8MS **(**1 **<<** 7**)**

; mini\_io status register

#define IO\_CSR\_RXNE **(**1 **<<** 0**)**

#define IO\_CSR\_TXE **(**1 **<<** 1**)**

#define IO\_CSR\_TI **(**1 **<<** 2**)**

#define IO\_CSR\_BTN **(**1 **<<** 7**)**

; Mass storage commands

#define IO\_MS\_CMD\_RD 1

#define IO\_MS\_CMD\_WR 2

#define IO\_MS\_CMD\_SYNC 3

;=====================================

**Listing 2. Program demonstracyjny dla mikroprocesora 8085**

; SDC\_One 8085 demo

; gbm 02’2018

; assemble: tasm -85 -x3 sdcdemo\_85.asm sdcdemo\_85.hex

#include “miniio.h”

.org 0

start**:**

mvi a**,** IO\_TMR\_X8MS **|** 125 ; 125 \* 8 ms = 1s

**out** IO\_TMR ; setup timer

wait1s**:**

**in** IO\_CSR ; get timer status

ani IO\_CSR\_TI

**jz** wait1s

**out** IO\_CSR ; clear timer flag

**in** IO\_LEDS ; get LED state

adi ‘A’ ; convert to letter - increment

**mov** b**,** a  ; save letter in B

ani 0fh

**out** IO\_LEDS ; set LEDs

**in** IO\_CSR ; get console status

ani IO\_CSR\_TXE

**jz** wait1s

**mov** a**,** b

**out** IO\_CON ; output a letter

**jmp** wait1s

.end

**Listing 3. Program demonstracyjny dla mikroprocesora Z80CPU**

; SDC\_One Z80CPU demo

; gbm 02’2018

; assemble: tasm -80 sdcdemo\_z80.asm sdcdemo\_z80.hex

#include “miniio.h”

.org 0

start**:**

ld a**,** IO\_TMR\_X8MS **|** 125 ; 125 \* 8 ms = 1s

**out** **(**IO\_TMR**),** a  ; setup timer

wait1s**:**

**in** a**,** **(**IO\_CSR**)** ; get timer status

**and** IO\_CSR\_TI

jr z**,** wait1s

**out** **(**IO\_CSR**),** a  ; clear timer flag

**in** a**,** **(**IO\_LEDS**)** ; get LED state

**add** a**,** ‘A’ ; convert to letter - increment

ld b**,** a  ; save letter in B

**and** 0fh

**out** **(**IO\_LEDS**),** a  ; set LEDs

**in** a**,** **(**IO\_CSR**)** ; get console status

**and** IO\_CSR\_TXE

jr z**,** wait1s

ld a**,** b

**out** **(**IO\_CON**),** a  ; output a letter

jr wait1s

.end

**Listing 4. Program demonstracyjny dla mikroprocesora W65C02S**

; SDC\_One 65c02 demo

; gbm 02’2018

; assemble: tasm -65 -x3 sdcdemo\_65.asm sdcdemo\_65.hex

#define MINIIO\_BASE 0fe00h

#include “miniio.h”

.org 0ff00h

start**:**

lda #IO\_TMR\_X8MS **|** 125 ; 125 \* 8 ms = 1s

sta IO\_TMR ; setup timer

wait1s**:**

lda IO\_CSR ; get timer status

**and** #IO\_CSR\_TI

beq wait1s

sta IO\_CSR ; clear timer flag

lda IO\_LEDS ; get LED state

**clc**

**adc** #’A’ ; convert to letter - increment

tax ; save letter in X

**and** #$0f

sta IO\_LEDS ; set LEDs

lda IO\_CSR ; get console status

**and** #IO\_CSR\_TXE

beq wait1s

stx IO\_CON ; output a letter

bra wait1s

; reset vector

.org 0fffch

.dw start

.end

**Listing 5. Program demonstracyjny dla mikroprocesora MC68008**

; SDC\_One MC68008 demo

; gbm 02'2018

; assemble with Easy68K

;========================================

MINIIO\_BASE equ **-**256

; MINI IO registers

IO\_LEDS equ MINIIO\_BASE**+**0

IO\_CON equ MINIIO\_BASE**+**1

IO\_TMR equ MINIIO\_BASE**+**2

IO\_CSR equ MINIIO\_BASE**+**3

IO\_TMR\_X8MS equ 1**<<**7

IO\_CSR\_RXNE equ 1**<<**0

IO\_CSR\_TXE equ 1**<<**1

IO\_CSR\_TI equ 1**<<**2

;========================================

Org 0

dc.l $1000 ; initial SSP value

dc.l start ; program entry point

org $400

start**:**

move.b #**(**IO\_TMR\_X8MS **|** 125**),** IO\_TMR ; 125 \* 8 ms = 1s

wait1s**:**

btst.b #2**,** IO\_CSR ; get timer status

beq.b wait1s

move.b #IO\_CSR\_TI**,** IO\_CSR

move.b IO\_LEDS**,** d0 ; get LED state

add.b #'A'**,** d0 ; convert to letter - increment

move.b d0**,** d1 ; save letter

andi.b #$0f**,** d0

move.b d0**,** IO\_LEDS ; set LEDs

btst.b #1**,** IO\_CSR ; get console status

beq.b wait1s

move.b d1**,** IO\_CON ; output

bra.b wait1s

end