

Laboratorium

Programowania Kart Elektronicznych

Programowanie BasicCard

Marek Goślawski

Programowanie BasicCard

- Przygotowanie do zajęć
 - dokumentacja ZeitControl BasicCard
 - środowisko programistyczne
 - karta BasicCard
- Potrzebne wiadomości
 - język angielski w stopniu pozwalającym na czytanie dokumentacji technicznej
 - składnia poleceń i odpowiedzi APDU
 - podstawowa znajomość składni języka Basic

Programowanie BasicCard

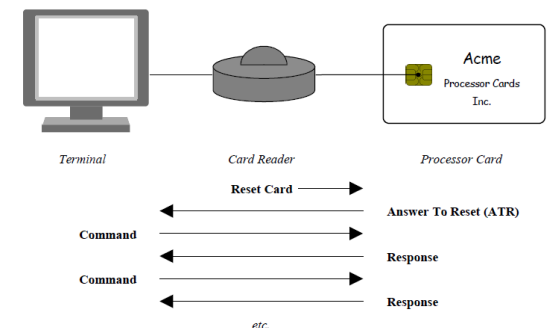
- Oprogramowanie

- środowisko ZeitControl

- Professional Development Environment (zcpde.exe)
 - ZC-Basic Compiler (ZCMBasic.exe)
 - BasicCard Program Debugger (ZCMDCard.exe)
 - Terminal Program Debugger (ZCMDTerm.exe)
 - MultiSimulator (ZCMSIM.EXE)

- program

- karta
 - terminal



Programowanie BasicCard

- Rodzaje plików
 - aplikacje (*.BAS)
 - obrazy (*.IMG)
 - pliki debuggera (*.DBG)
 - pliki programów karty (*.ZCC)

Programowanie BasicCard

- Stany karty
 - NEW
 - LOAD, PERS, TEST
 - RUN
- między stanami LOAD, PERS, TEST można przechodzić dowolną liczbę razy
- **po przejściu do stanu RUN nie można przejść do innego stanu!**

Programowanie BasicCard

- Rodzaje zmiennych
 - eeprom
 - pamięć stała, odpowiednik „twardego dysku”,
 - dostępna globalnie
(dostępna dla każdej procedury w programie)
 - public
 - pamięć ulotna (RAM), dostępna kiedy karta jest zasilana,
 - dostępna globalnie
(dostępna dla każdej procedury w programie),

Programowanie BasicCard

- Rodzaje zmiennych

- static

- pamięć ulotna (RAM), dostępna kiedy karta jest zasilana,
 - dostępna lokalnie (dostępna dla procedury w której została zadeklarowana),

- private

- pamięć ulotna (RAM), dostępna w czasie wykonywania procedury, w której została zadeklarowana,
 - dostępna lokalnie (dostępna dla procedury w której została zadeklarowana)

Programowanie BasicCard

- Elementy języka Basic

- typy danych

- Byte, Integer, Long, Single, String, String*n
 - Tablice

- deklaracje danych

```
Eeprom S1 As String*5 = "ABC" ' Padded with two NULL bytes
```

```
Public S2 As String*3 = &H81, &H82, &H83
```

```
Private S3 As String*7 = 3, 4, "XYZ"
```

```
Dim A1()
```

```
Rem This is equivalent to:
```

```
Rem Private S3 As String*7 = 3, 4, 88, 89, 90, 0, 0
```


Programowanie BasicCard

- Elementy języka Basic

- wyrażenia

`X + 5 ' Apply '+' (addition) to terms X and 5`

`A(I) * Rnd ' Apply '*' (multiplication) to terms A(I) and Rnd`

`S$ + "0" ' Apply '+' (concatenation) to terms S$ and "0"`

- przypisania

`Let Y = X + 5`

- polecenia kontrolujące działanie programu

`Exit`

`label1: GoTo label1`

`label2: GoSub label2`

`If X = 0 Then GoTo 100`

`If X = 0 Then Y = 0 : If Z = 0 Then GoTo 100 ' Can be nested`

`If X < 0 Then`

`X = 0`

`ElseIf X > 50 Then X = 50`

Programowanie BasicCard

- Elementy języka Basic
 - polecenia kontrolujące działanie programu

```
For I = 1 To 2 : For J = 1 To 2  
Print 10*I + J : Exit For  
Next I, J
```

```
While ... Wend  
Do ... Loop  
Select Case ... End Select  
On ... GoTo ...
```

Programowanie BasicCard

- Elementy języka Basic
 - procedury, funkcje, polecenia, blok inicjujący (poza definicją procedury, funkcji lub polecenia)

```
Function Square (X!) As Single
```

```
    Square = X! * X!
```

```
End Function
```

```
Sub S()
```

```
    Y! = Square (5.5)
```

```
End Sub
```

```
Command &H20 &H00 HelloWorld (ReturnValue$)
```

Programowanie BasicCard

- Elementy języka Basic
 - polecenia specyficzne dla Terminala

`Cls`

`Print`

`InKey$ ...`

`CardReader, CardInReader, ResetCard`

`WriteEeprom`

`...`

Programowanie BasicCard

- Przykłady

- program karty

```
Declare ApplicationID = "HelloWorld"
Public SHelloWorld As String = "Hello World!,,
Command &H20 &H00 HelloWorld (ReturnValue$)
    ReturnValue$ = SHelloWorld
End Command
```

- program terminala

```
#Include CARDUTIL.DEF
#Include COMMERR.DEF
Declare Command &H20 &H00 HelloWorld (ReturnValue$)
Public S$ as String
Call WaitForCard()
ResetCard : Call CheckSW1SW2()
Call HelloWorld(S$) : Call CheckSW1SW2()
Print S$
```

Programowanie BasicCard

- Zadania
 - program „Hello <student>” (3 warianty)
 - symulator (program i screeny)
 - karta (logi komunikacji z kartą)
 - BalanceReader (program + foto, ew. demonstracja)
- przesłanie na adres:
 - marek.goslowski@put.poznan.pl

Zadanie 1: „Hello <student>” – symulator

Zadanie 2: „Hello <student>” – karta

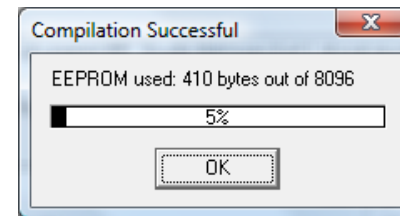
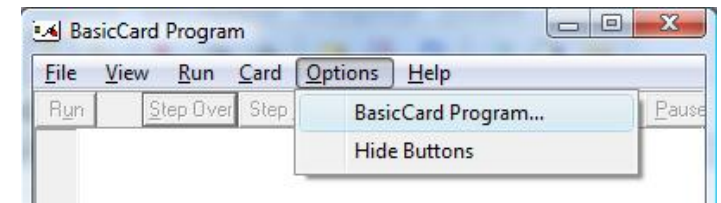
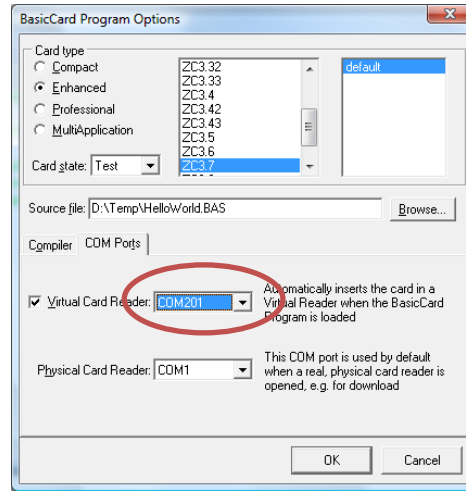
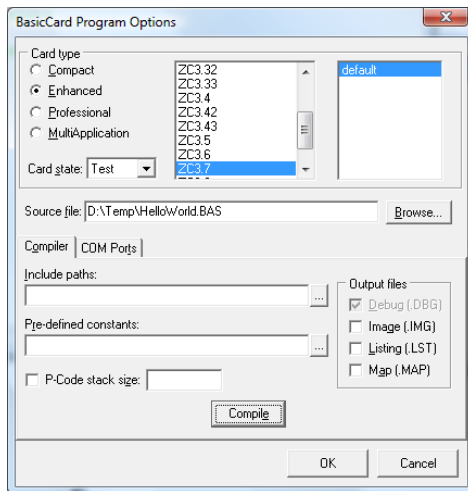
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:

marek.goslowski@put.poznan.pl

Programowanie BasicCard

- Uruchomienie środowiska karty (ZCMDCard.exe)
 - kompilacja

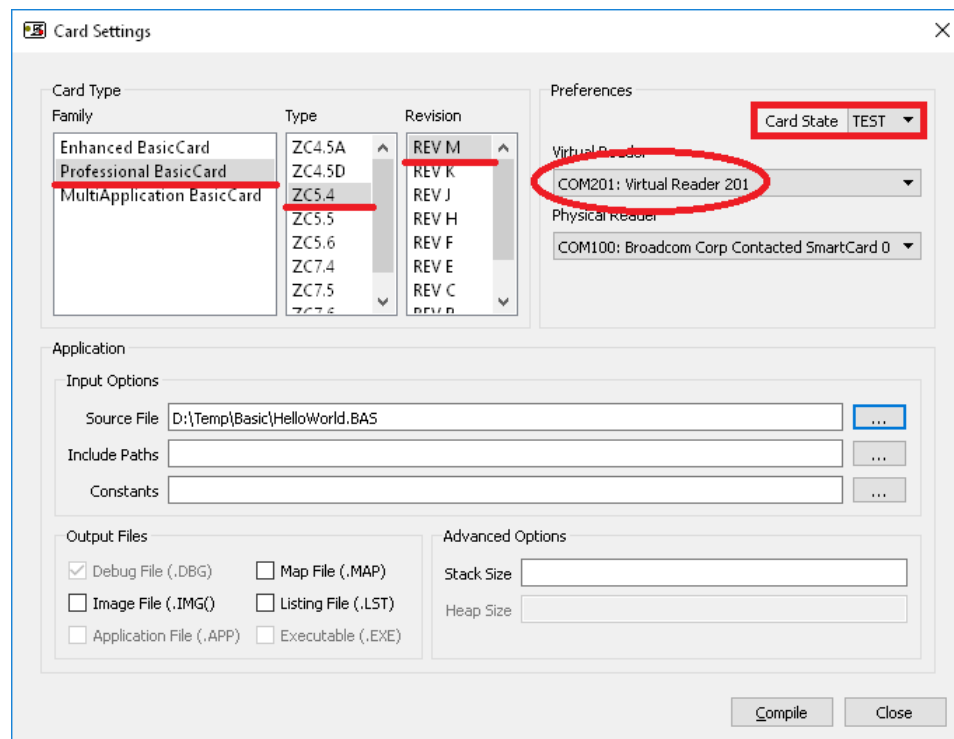
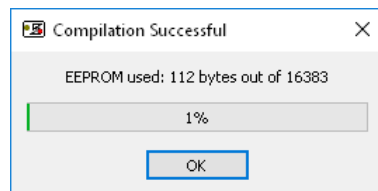
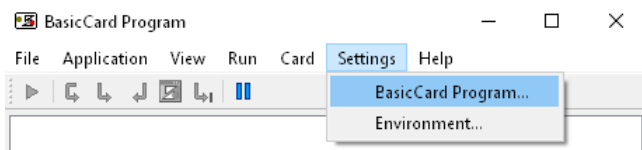


- Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

- Uruchomienie środowiska karty v8 (ZCMDCard.exe) – kompilacja

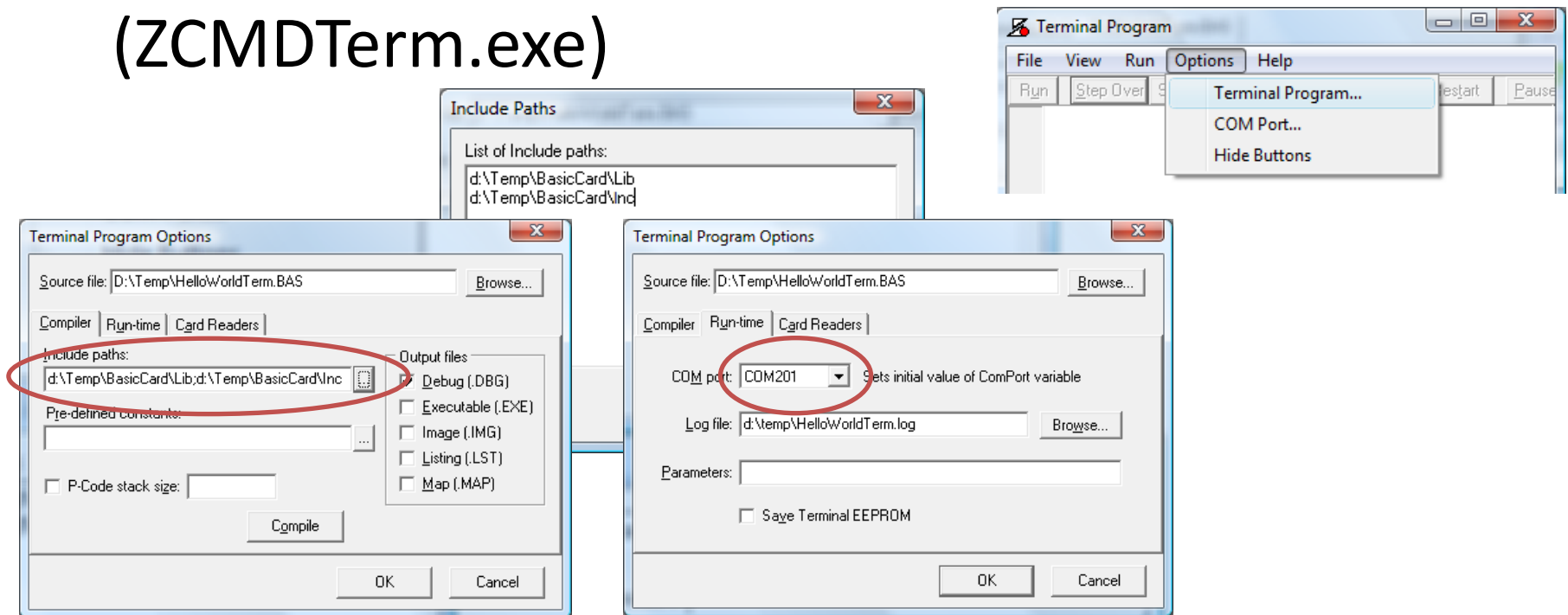


- Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

- Uruchomienie środowiska terminala (ZCMDTerm.exe)

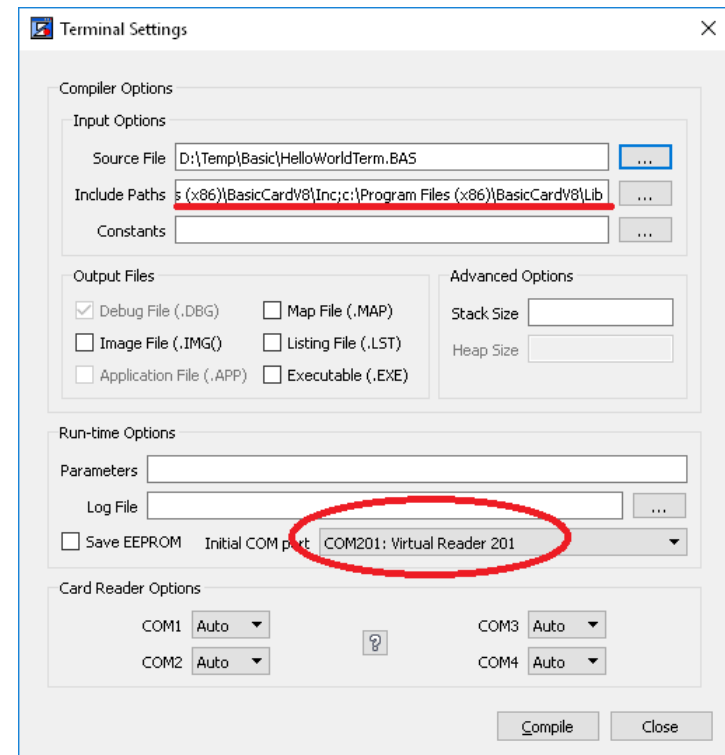
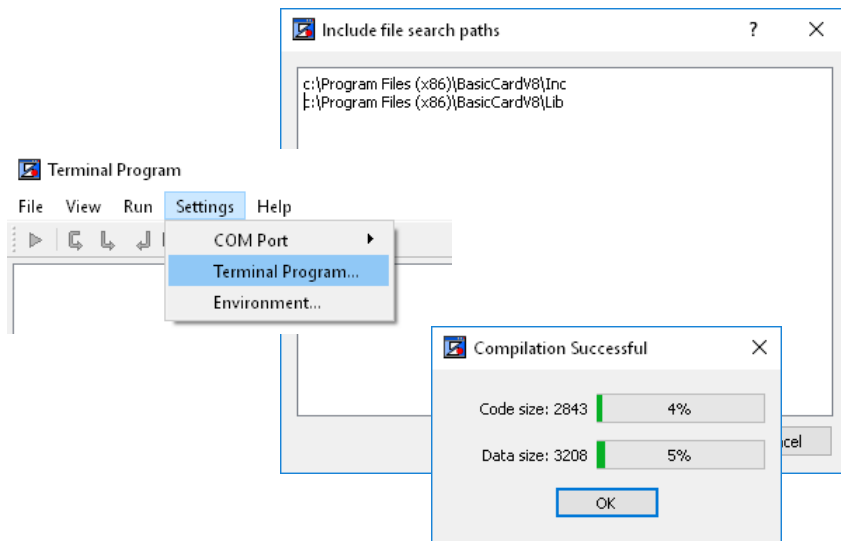


- Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

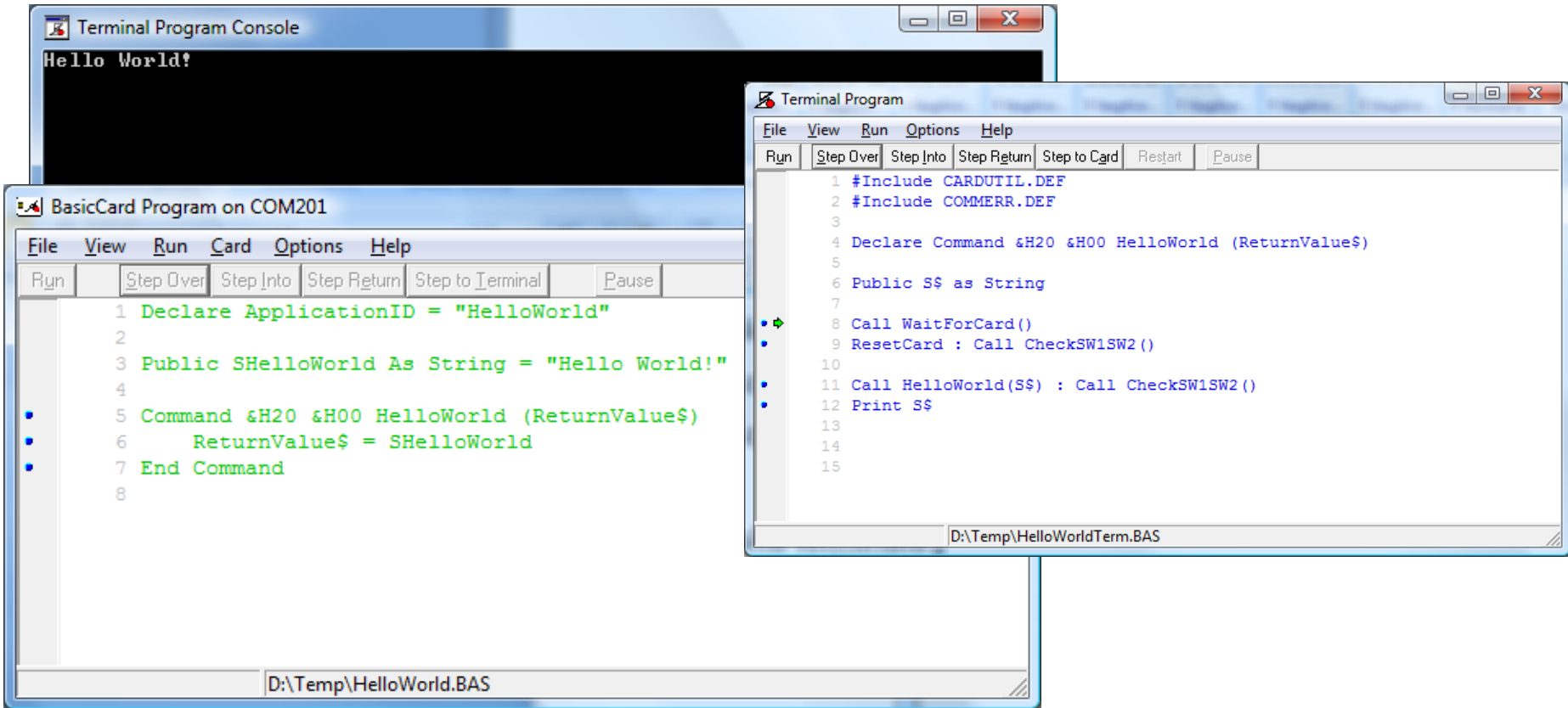
- Uruchomienie środowiska terminala v8 (ZCMDTerm.exe)



Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

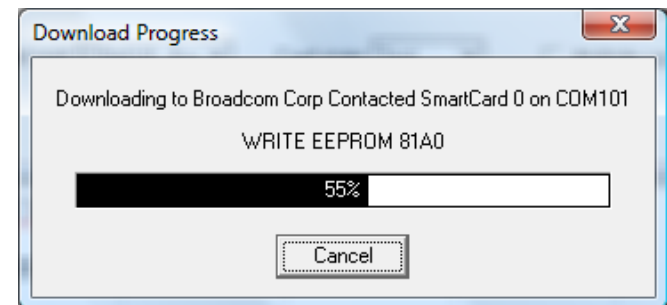
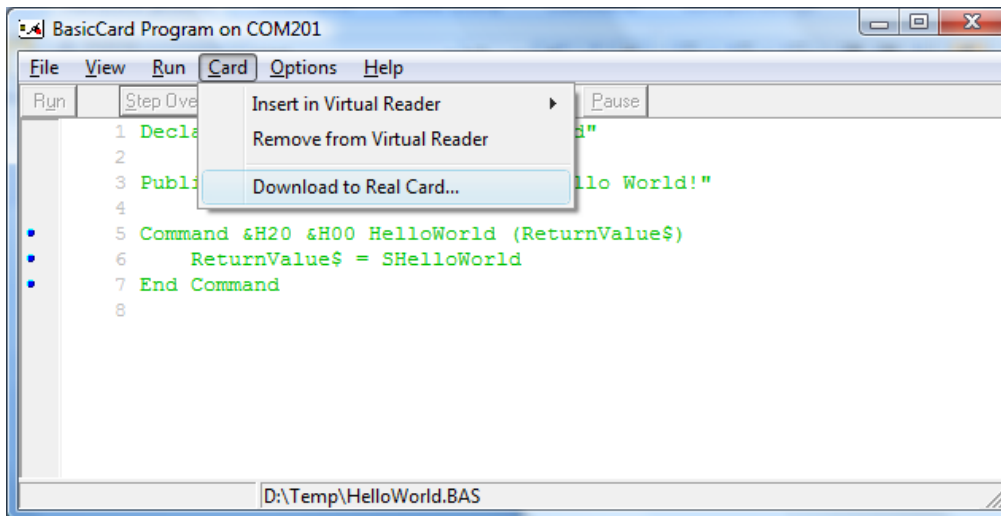


- Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

- Załadowanie programu do pamięci karty
 - *.img lub *.dbg
 - kompilacja dla właściwej wersji karty!

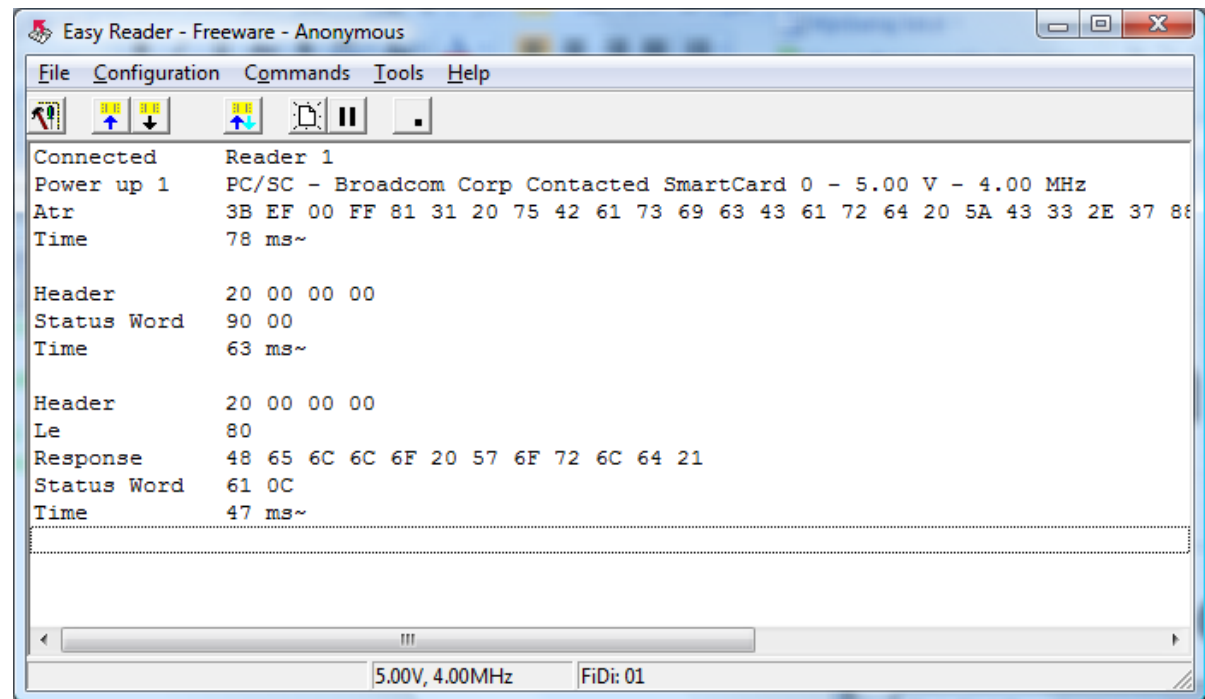
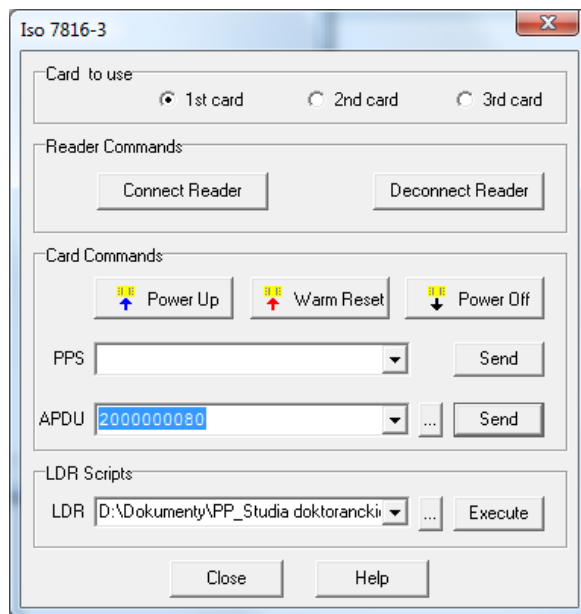


Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

- Sprawdzenie działania programu na karcie



- Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

- BalanceReader

- `command &HC8 &H00 PRDisplay ...`



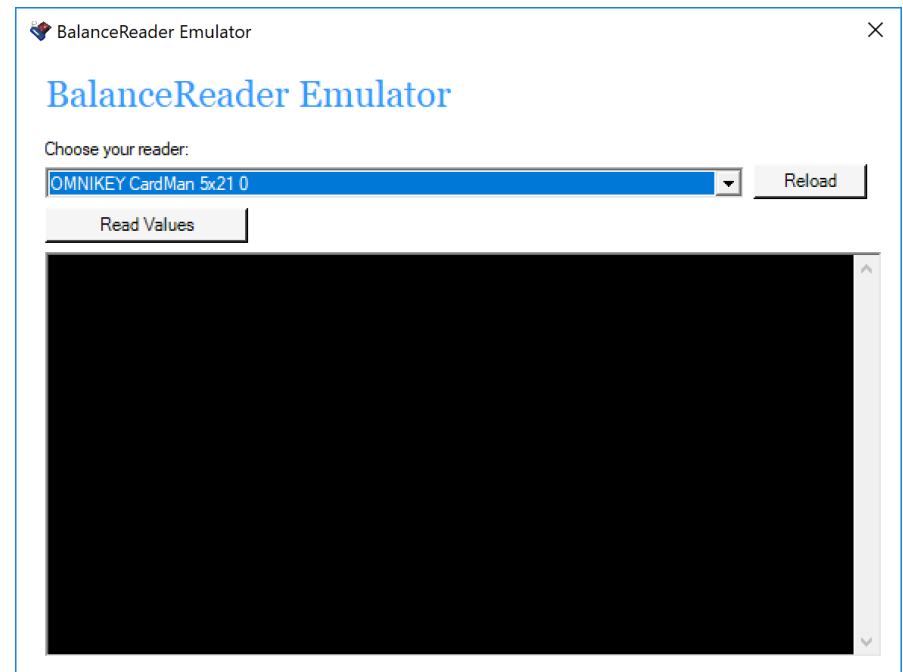
źródło: <http://www.basiccard.com/>

- Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

Programowanie BasicCard

- BalanceReader
 - emulator 😊



Zadanie 1: „Hello <student>” – symulator
Zadanie 2: „Hello <student>” – karta
Zadanie 3: „Hello <student>” – BalanceReader

wysłać na adres:
marek.goslowski@put.poznan.pl

- marek.goslowski@put.poznan.pl
- +48 61 665 3680
- +48 694 949 750
- pl. Marii Skłodowskiej-Curie 5 (Wilda)
Budynek B1 (Rektorat), pok. 405
- <http://mcp.poznan.pl/>