|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| logowydzialu | Instytut Informatyki Politechniki Śląskiej  Zespół Mikroinformatyki i Teorii Automatów Cyfrowych  **Laboratorium SMiW** | | logoii | |
| **Rok akademicki** | **Rodzaj studiów\*: SSI/NSI/NSM** | **Numer ćwiczenia:** | **Grupa** | **Sekcja** |
| **2015/2016** | **SSI** | **18** | **5** | **3** |
| **Data i godzina planowana ćwiczenia:**  dd/mm/rrrr - gg:mm | **20/10/2015 - 13:15** | **Prowadzący**:  OA/JP/KT/GD/BSz/GB | **JP** | |
| **Data i godzina wykonania ćwiczenia:**  dd/mm/rrrr - gg:mm | **20/10/2015 - 13:15** |
| ***Sprawozdanie*** | | | | |
| **Temat ćwiczenia:**  Mikrokontrolery serii AVR | | | | |
| **Skład sekcji:** | 1. Michał Lytek  2. Jakub Świerczek | | | |

# Zadanie

Pamięć ROM procesu ATmega2560 ma niewiadomy rozmiar. Koniec tablicy sygnalizowana jest dwoma zerami koło siebie.

Napić program w assemblerze oraz w C, który po naciśnięciu przycisku dowolnego przycisku, które są podłączone do portu A (Przycisk nie wciśnięty – stan wysoki), wyświetli na diodach (Port B – stan wysoki – dioda świeci) kolejne komórki z pamięci ROM.

# Wykonanie

Nasz program czeka na naciśniecie dowolnego przycisku i wtedy wyświetla na diodach kolejne bajty z ROM. Jeżeli nie jest naciśnięty żadne przycisk to nie wyświetla, gdy znów naciśniemy przycisk pamięć jest wyświetlana od następnego elementu tablicy. Jeżeli dojdziemy do końca pamięci to pierwsze zera przepisywane są na diody aby je zgasić a następnie program się kończy.

# Kod C i ASM

//TODO wstawić dobre kody ;)

# Testowanie i Uruchamianie

Testowanie polegało na ustawieniu pamięci ROM przed uruchomieniem programu. Podczas debuggowania ustawialiśmy Port A imitując przyciskanie przycisków. Następnie obserwując kolejne komórki pamięci sprawdzamy czy dobrze są wyświetlanie na Portcie B.

Nie porównywaliśmy czasów wykonania poszczególnych programów, ponieważ podczas debuggowania ustawiamy Port A i ciężko by było zmierzyć czas musimy sami „wciskać przyciski”

# Wnioski