VI Konferencja

eTechnologie w Kształceniu Inżynierów eTEE'2019

Politechnika Gdańska

LABORATORYJNY SYSTEM CYFROWY PROGRAMOWANY PRZEZ ETHERNET OPARTY NA MAGISTRALI SPI

Krystyna Maria NOGA
Dorota RABCZUK

19-20 wrzesień 2019

Plan

- Wstęp
- Topologia systemu cyfrowego opartego na magistrali SPI
- Kontrola sytemu cyfrowego przez Ethernet - etapy rozbudowy sprzętowej i programowej projektu
- Projektowanie algorytmu dla układu programowalnego FPGA
- Podsumowanie - walory dydaktyczne projektu

eTEE
e-Technologies
Engineering Education

Wstęp

- Układy cyfrowe wykorzystywane w systemach wbudowanych i Internetu rzeczy są powszechnie wyposażane w co najmniej jeden interfejs szeregowy, np. **SPI** (**Serial Peripheral Interface**), takie rozwiązanie ułatwia przyłączenie urządzeń cyfrowych do systemu.
- Rynek elektroniczny oferuje szeroką gamę **urządzeń sterowanych po SPI**, w tym: **potencjometry, pamięci, wyświetlacze7-segmentowe**, wyświetlacz LCD, **różne czujniki, adapter Ethernet.**
- Algorytm magistrali SPI można zaimplementować w układzie programowalnym **CPLD** lub **FPGA**, który może pracować w jednym systemie z **µk** i czujnikami cyfrowymi.

- Układ programowalny oraz **µk** powinny pełnić różne funkcje w projektowanym systemie:
  - układ CPLD lub FPGA, ze względu na szybkie przetwarzanie równoległe sygnałów, powinien wykonywać **algorytmy obliczeniowe**,
  - **µk**, wyposażony standardowo w wiele interfejsów, jest przystosowany do **kontroli czujników** i **prowadzenia transmisji** danych po magistralach systemowych.

- Master **µk**, za pośrednictwem adaptera Ethernetu, utrzymuje kontakt z odległym stanowiskiem użytkownika utworzonym na komputerze PC. Pozwala to na wykorzystanie w dydaktyce systemu zdalnie sterowanego.

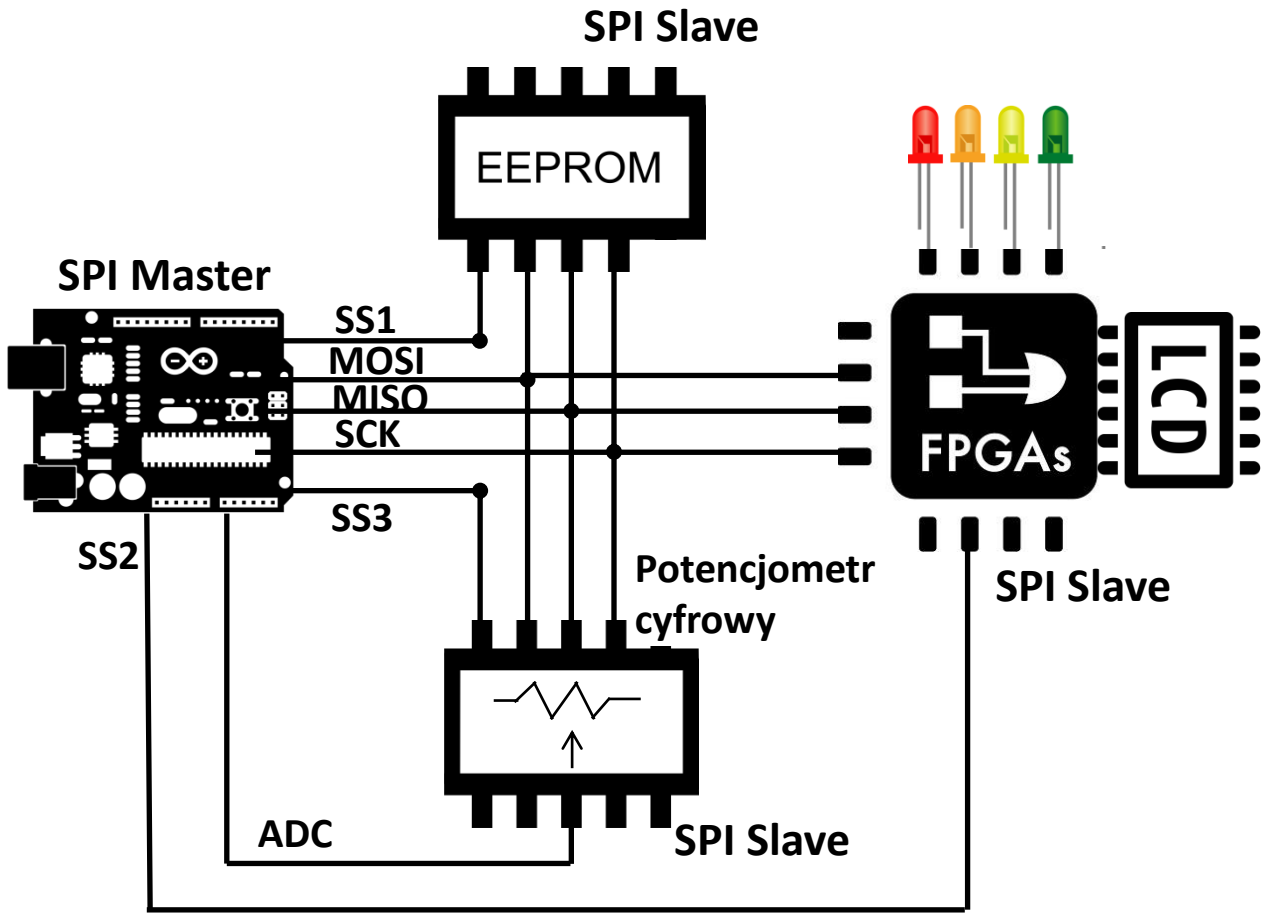
Topologia systemu cyfrowego opartego na magistrali SPI

- W prezentowanym systemie cyfrowym µk pełni na magistrali SPI rolę urządzenia **Master**.
- **Master** SPI jest połączony z urządzeniami **Slave** SPI 4 liniami:

❖ **MOSI, MISO, SCK wspólnymi** dla wszystkich urządzeń na magistrali,

❖ **SS** (**Slave Select**) prowadzonymi oddzielnie od µk Master do każdego urządzenia Slave, stanowi linię wyboru Slave'a.

- Master dokonuje wyboru urządzenia Slave do dwukierunkowej transmisji danych wystawiając stan niski na jego linii wyboru i utrzymując go przez cały czas transmisji.
- Transmisja na magistrali SPI ma charakter dookólny, a pracujące na niej urządzenia zachowują się jak rejestry przesuwne.
- Na załączonym schemacie **linia ADC** nie należy do magistrali SPI. Linia ta została podłączona z jednej strony do analogowego suwaka potencjometru sterowanego cyfrowo, a z drugiej do wejścia przetwornika analogowo-cyfrowego µk umożliwiając odczyt napięcia na suwaku.



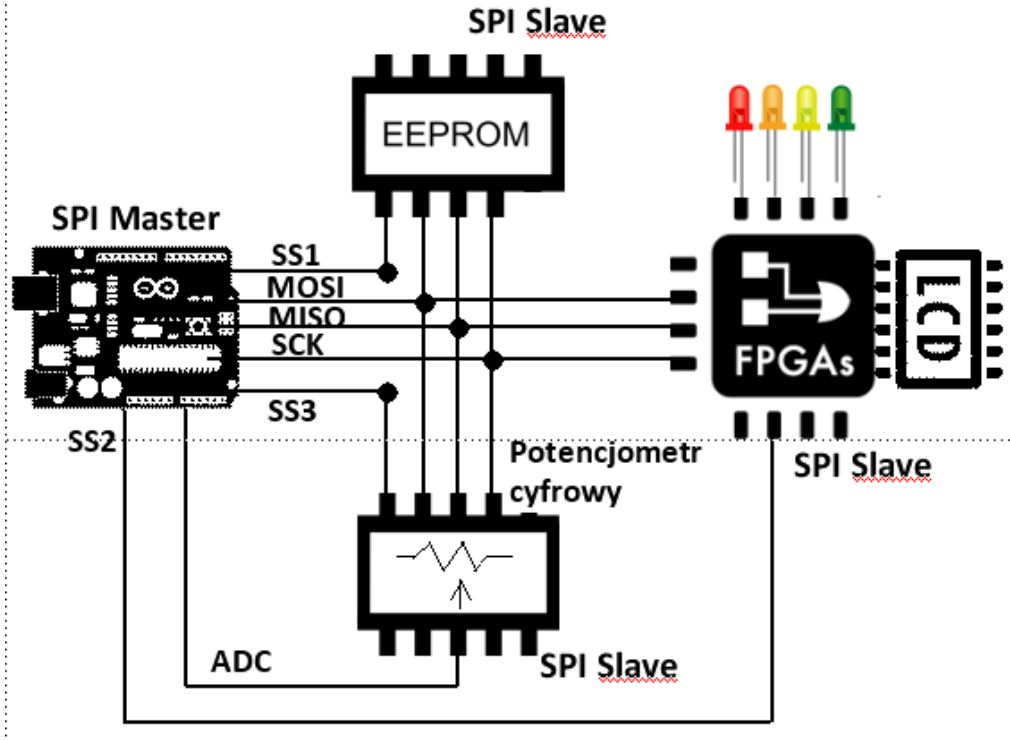
MOSI - **Master Output Slave Input**,
MISO - **Master Input Slave Output**,
SCK – **Clock linia zegarowa synchronizująca transmisję danych**,
SS1, SS2, SS3 – **Slave Select 1, 2, 3.**

Topologia systemu cyfrowego opartego na magistrali SPI cd.

- Do pełnienia roli **Master** może zostać skonfigurowany **µk** lub układ **FPGA** – pozostałe mikrokontrolery i układy FPGA muszą pełnić rolę Slave'ów.

- W prezentowanym projekcie rolę **Master** pełni **µk** . Wybór jest uzasadniony tym, że użyty do projektu układ FPGA nie ma interfejsu Ethernet, a **zdalne sterowanie po sieci Ethernet** jest założeniem systemu.

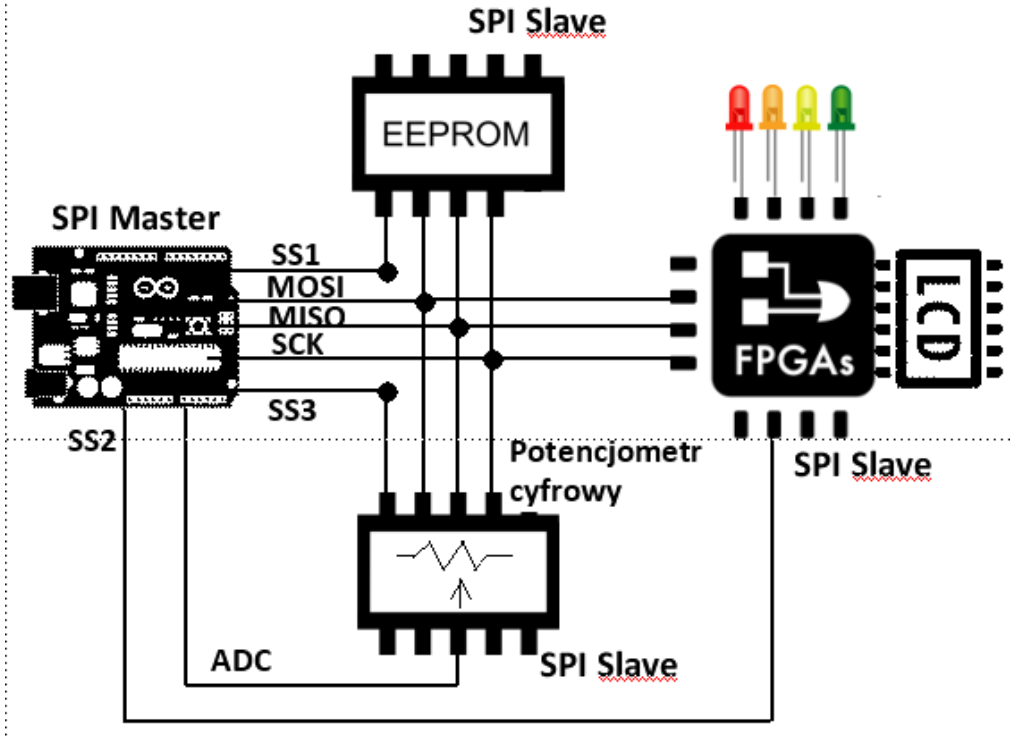
- **µk** współpracuje z zewnętrznym adapterem Ethernet, **który jest z nim połączony również po magistrali SPI** – układu tego nie uwidoczniono na wyświetlanym schemacie (ale znajduje się na schemacie topologii w wydrukowanym artykule).



@

Topologia systemu cyfrowego opartego na magistrali SPI cd.

- Dzięki popularności **magistrali SPI** w projekcie można wykorzystać szeroką gamę różnych układów cyfrowych wyposażonych w tę magistralę.
- Podczas prac nad projektem poszczególne grupy studenckie pisały i implementowały oprogramowanie uwzględniające obsługę:
  - zewnętrznej nieulotnej pamięci EEPROM,
  - cyfrowego zegara czasu rzeczywistego,
  - termometru cyfrowego,
  - monitora LCD,
  - potencjometru cyfrowego.



- Oprogramowanie dla wymienionych urządzeń jest opracowane w **języku C** i kompilowane w **środowisku Atmel Studio** lub **Arduino IDE (Integrated Development Environment)** .
- Liczne biblioteki dostępne na licencji freeware ułatwiają tworzenie oprogramowania.
- Odrębnym zadaniem programistycznym jest napisanie programu komunikacyjnego dla FPGA tworzonego w języku VHDL lub Verilog w środowisku **Quartus** lub **Max Plus Baseline**.

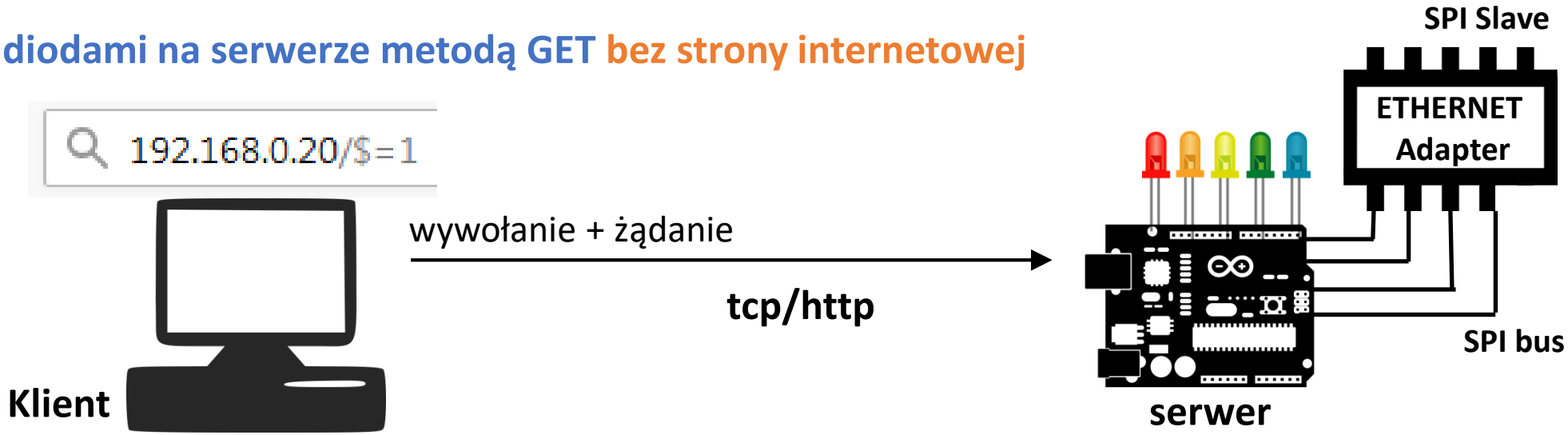
@

Kontrola systemu cyfrowego przez Ethernet – etapy rozbudowy sprzętowej i programowej projektu

Założenia:

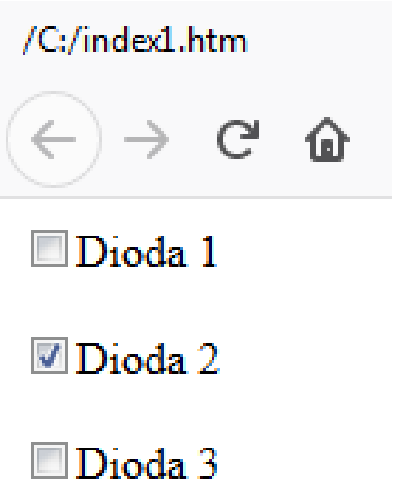
- Na µk powstaje **serwer** tcp/http, do którego klient sieci tcp zgłasza się przez przeglądarkę.
- Do sterowania po sieci Ethernet wykorzystuje się metodę http GET (podtrzymywana przez biblioteki mikrokontrolerów).
- **Najprostszy program sterujący nie posiada strony internetowej – klient wpisuje rozkazy w linii poleceń przeglądarki.**
- **W bardziej rozbudowanych programach strona internetowa jest przechowywana w pamięci SD połączonej z µk po magistrali SPI (pominięto na schematach).**
- Utworzenie strony internetowej w HTMLu jest elementem programu i ćwiczenień dla studentów.
- W komunikacji po sieci Ethernet wykorzystano obiekt **XMLHttpRequest** oraz interfejs AJAX **(Asynchronous Java Script and XML)** - do sprawnej modyfikacji stanów obiektów na stronie internetowej.

Sterowanie diodami na serwerze metodą GET bez strony internetowej



Żądanie http (w tym przykładzie łańcuch „$=1”) jest dopisywane bezpośrednio do adresu serwera w linii poleceń przeglądarki

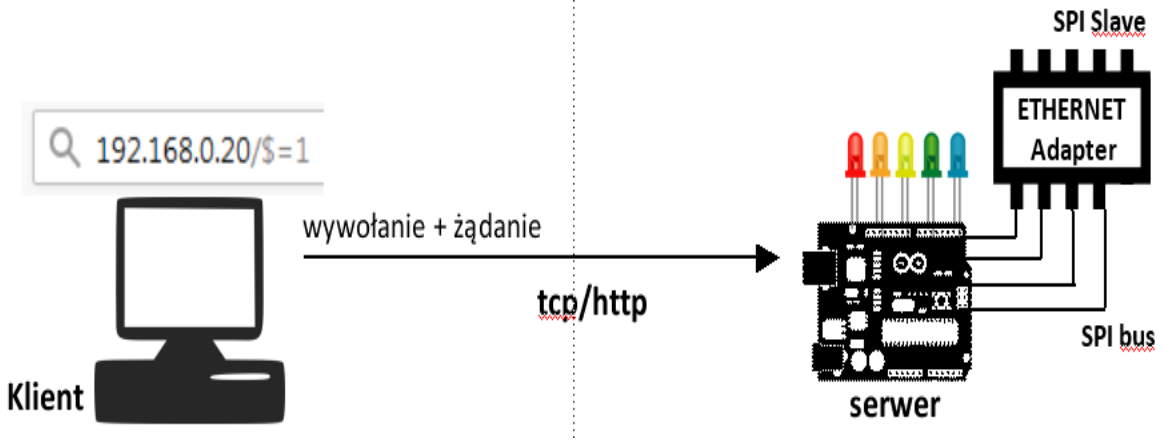
Sterowania diodami na serwerze metodą GET przez stronę internetową



- Założono stronę internetową w języku **HTML** z komponentami reprezentującymi diody (checkboxy).
- Metoda GET jest wywoływana cyklicznie przez interfejs programowania aplikacji **AJAX (Asynchronous Java Script and XML)** , co umożliwia zmianę stanu wybranych komponentów strony internetowej asynchronicznie, bez konieczności przeładowywania całej strony.
- *request.open("GET", "ajax" + strDioda1 + strDioda2 + strDioda3 + nocache, true);*
- Stany checkboxów reprezentujących diody są przełączane „a priori” w momencie kliknięcia.

Sterowanie

- Założeniem projektu jest sterowanie zdalne po sieci **Ethernet** realizowane z wykorzystaniem gotowej biblioteki Ethernet dostępnej na licencji **freeware**.



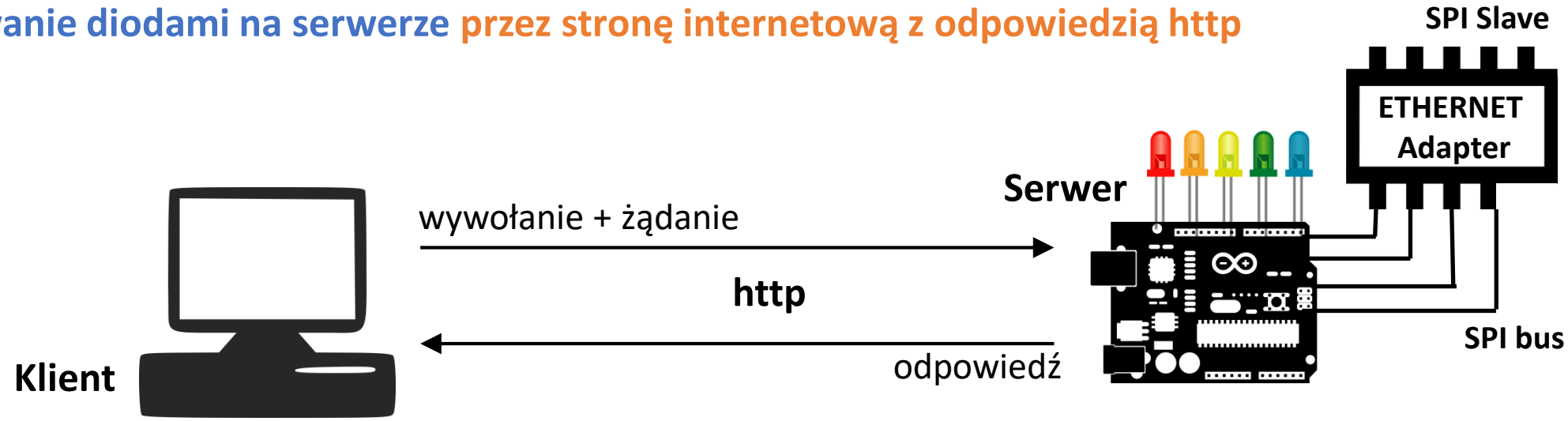
- Dalszy rozwój projektu przewiduje pisanie w **języku HTML** i wizualne prezentowanie urządzeń na serwerze w postaci komponentów formularza **HTML**.

- Liczba podstawowych komponentów formularza do wyboru jest niewielka (np. **checkbox, radiobutton, textbox**) stąd stworzona w ten sposób strona internetowa jest niezbyt atrakcyjna wizualnie, ale nie o to chodzi w realizowanym projekcie.

- Zadaniem studenta jest zastosowanie techniki **AJAX** (Asynchronous Java Script and XML) czyli **zbioru narzędzi asynchronicznych JavaScriptu** do **podmieniania na stronie webowej stanów urządzeń podpiętych do serwera**.

@

Sterowanie diodami na serwerze przez stronę internetową z odpowiedzią http



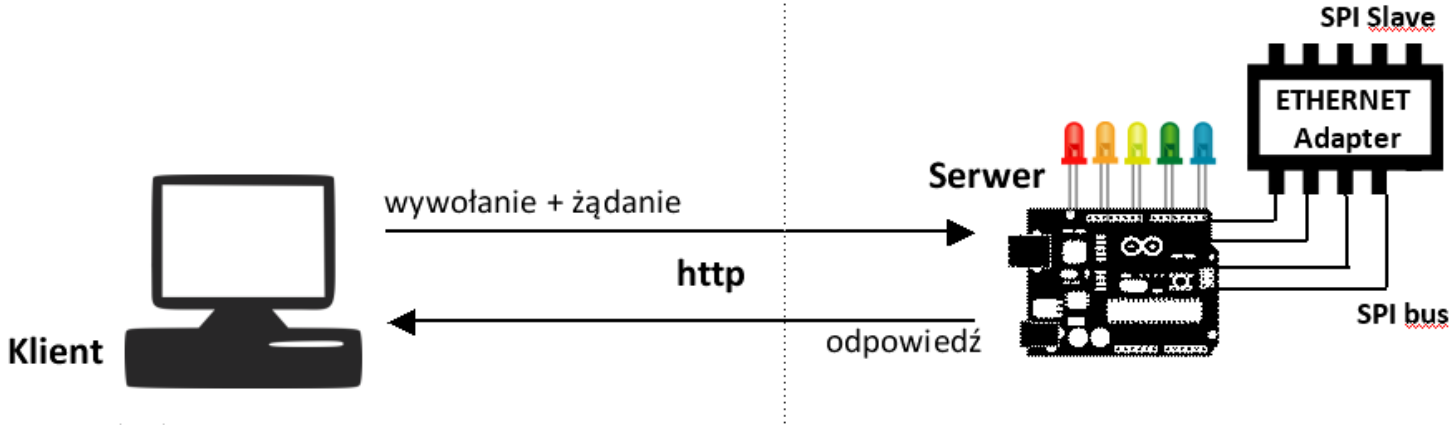
➢ **Wykorzystano obiekt XMLHttpRequest i jego metody do odebrania odpowiedzi od serwera z potwierdzeniem wykonania sterowania diodami.**

```
var request = new XMLHttpRequest();
request.onreadystatechange = function()
{
if (this.readyState == 4) {
        if (this.status == 200) {
        if (this.responseXML != null) {...}}}}
```

➢ Stany **checkboxów** reprezentujących diody nie są przełączane w momencie kliknięcia, lecz dopiero po odebraniu potwierdzenia.

- Na slajdzie pokazano fragment programu wykorzystującego **metodę** onready**statechange obiektu XMLHttpReuest**, która pozwala rozpoznać, że przyszła odpowiedź http od serwera.
- W dalszej części programu należy odczytać i rozpoznać łańcuchy tekstowe zawarte w tej odpowiedzi i następnie zobrazować stany na stronie webowej (tekstem lub na chechboxach).

Sterowanie na serwerze przez stronę internetową z odpowiedzią http - rozbudowa



- W ramach projektu można sformułować studentowi zadanie wykorzystania obiektów **DOM** (**Data Object Model**) języka HTML.

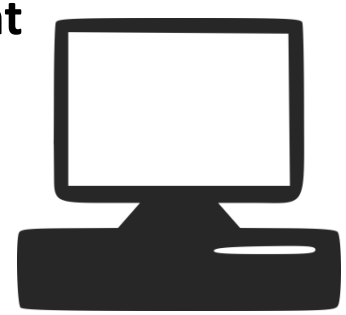
- Umożliwiają one odbiór odpowiedzi od serwera na przesłane mu żądanie http.

- Odpowiedź może zawierać **potwierdzenie wykonania polecenia** (w postaci dowolnego uzgodnionego łańcucha tekstowego) oraz **informacje o stanach urządzeń podłączonych na liniach serwera**.

- **Serwer jest urządzeniem Master** dla magistrali SPI, mamy dostęp do odczytu stanów urządzeń na magistrali SPI, np. temperatury, zawartości komórek pamięci EEPROM.

@

Sterowanie urządzeniami Slave na magistrali SPI serwera



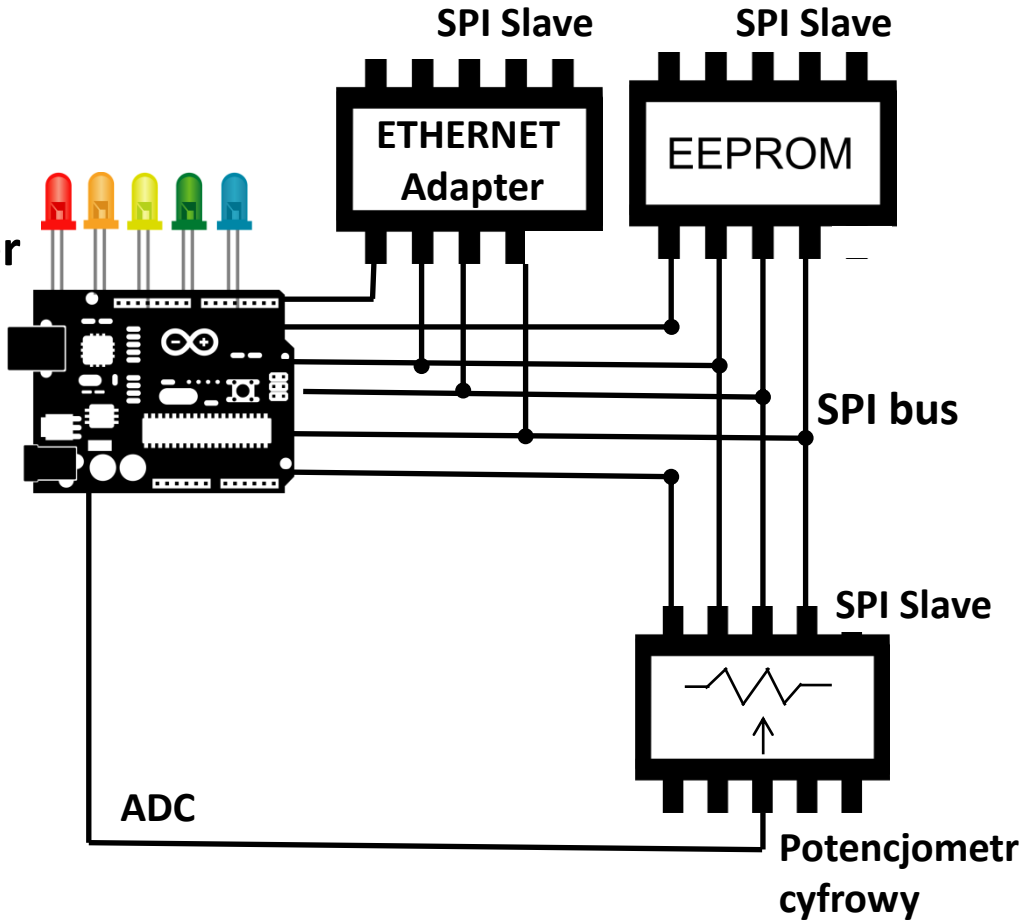
Klient

wywołanie + żądanie

http

odpowiedź

Serwer



- Do serwera podłączono urządzenia cyfrowe na magistrali SPI.
- Przygotowano program dla µk, który steruje urządzeniami oraz komponenty do wizualizacji na stronie internetowej.

Number of EEPROM Memory Cells to Write

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

Starting Cell

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

Select Slider Position

(15*10kOhm)/128
(16*10kOhm)/128
(17*10kOhm)/128
(18*10kOhm)/128

Potentiometer slider readback

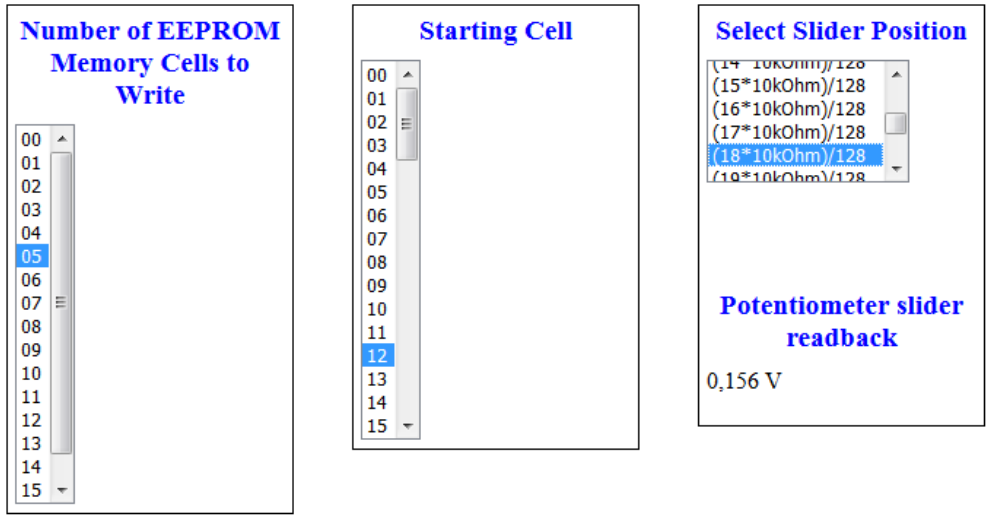
0,156 V

Data to memory

abcde

Readback from memory

abcde



*Widok prostej strony internetowej do zdalnego sterowania po sieci Ethernet z : pamięcią **EEPROM** oraz **potencjometrem cyfrowym**.*

Sterowanie urządzeniami Slave na magistrali SPI serwera

- Oprogramowanie obsługuje pamięć w zakresie **zapisu znaków,** podanych przez użytkownika, do komórek pamięci począwszy od komórki o podanym numerze.

- Pierwsza lista rozwijana pozwala wybrać **liczbę komórek**, które mają zostać zapisane,
- druga lista **numer pierwszej komórki** przeznaczonej do zapisania, w okienku tekstowym wpisuje się znaki do zapisania w pamięci (**tu: abcde**).

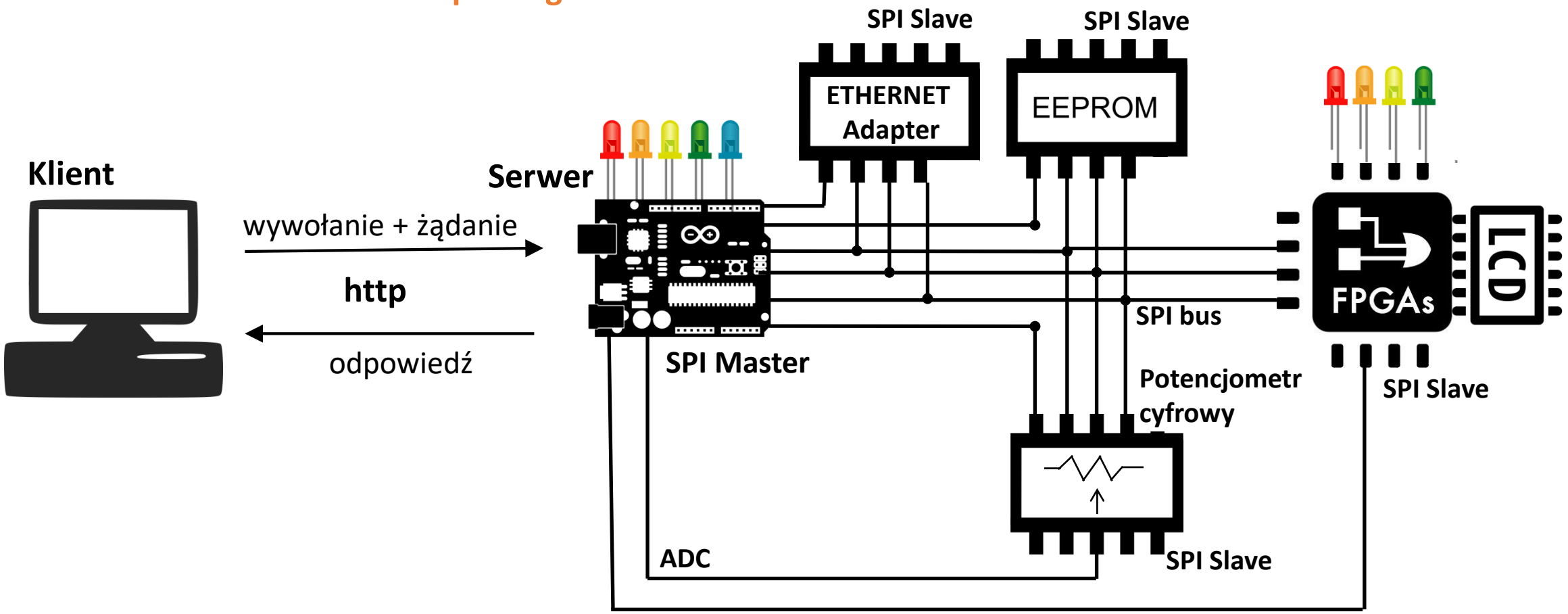
- Po wykonaniu rozkazu **µk Master** SPI **odczytuje** zwrotnie zapisane komórki i odsyła odczytane znaki w odpowiedzi http.
- Użytkownik może porównać znaki i upewnić się co do prawidłowości zapisu do pamięci.
- Rozkaz dla potencjometru zawarty w żądaniu http (w jednym żądaniu http są rozkazy dla wielu urządzeń) wskazuje, na jakiej pozycji ma być ustawiony suwak (w projekcie użyto potencjometr o **rozdzielczość siedmiobitowej** czyli o 128-miu pozycjach suwaka).

Sprawdzenie czy udało się ustawić suwak rozkazem cyfrowym wymaga odczytania napięcia na suwaku przez przetwornik ADC wbudowany w µk .

Odbywa się to poza magistralą SPI. Dla studenta jest to **dodatkowe zadanie** wymagające oprogramowania przetwornika ADC. Odczytana wartość napięcia jest odsyłana w odpowiedzi http i prezentowana wizualnie w postaci tekstowej (**tu: 0,156 V**).

@

Sterowanie układem FPGA po magistrali SPI



- **Układ FPGA został podłączony na magistrali SPI w charakterze urządzenia Slave i oprogramowany w języku VHDL.**
- Program realizuje dwa **procesy współbieżne**:
  - komunikacyjny magistrali SPI,
  - wyświetlania odebranych bajtów na LCD lub wyświetlaczach 7 - segmentowych.

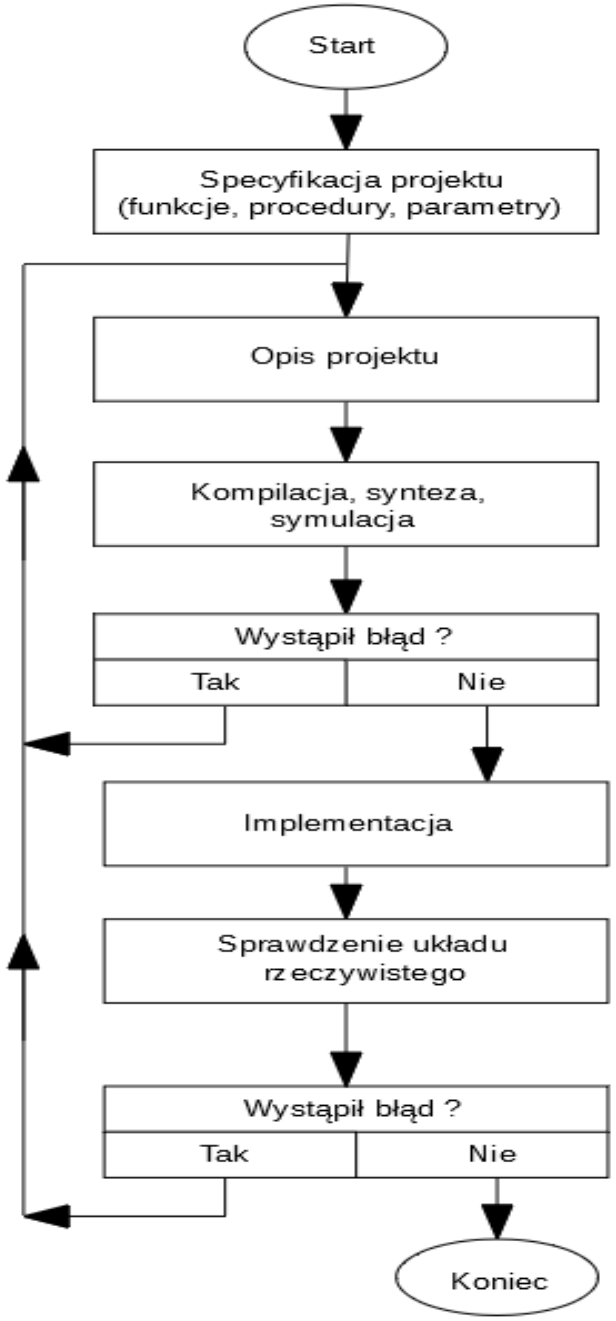
Projektowanie algorytmu dla układu programowalnego FPGA

- Algorytm komunikacji między **µk** a układem CPLD (FPGA) po magistrali SPI został zrealizowany w dwóch procesach.

- Proces odpowiedzialny za :
  - **transmisję danych** jest wykonywany w stanie niskim na linii wyboru SS,
  - **wyświetlenie odebranego bajtu** na diodach, wyświetlaczach 7 segmentowych lub LCD jest wykonywany w stanie wysokim na linii SS.

- **Projektowanie algorytmów** sterowania cyfrowego z wykorzystaniem języka opisu sprzętu **VHDL** (Very High Speed Integrated Circuits Hadrware Description Language) i ich **implementacja** w strukturach układów programowalnych (PLD) wymaga realizacji kilku etapów:
  - sporządzenia specyfikacji, czyli zdefiniowania niezbędnych funkcji, procedur, sygnałów wejściowych i wyjściowych,
  - opisu projektu, czyli zdefiniowanie wszystkich działań,
  - Po przeprowadzeniu kompilacji możliwa jest implementacja algorytmu w strukturze **PLD** oraz sprawdzenie układu rzeczywistego.

- Wśród pakietów oprogramowania narzędziowego dostępnych obecnie na rynku najczęściej wykorzystywany jest pakiet **Quartus** (wcześniejsza wersja to Max Plus II Baseline) firmy **Altera**, pakiet **Foundation ISE** i **WebPack** IDE firmy **Xilinx**. Pakiety te umożliwiają realizacje wszystkich etapów cyklu projektowego.



Projektowanie algorytmu dla układu programowalnego FPGA

- Do napisania kodu transmisji danych po SPI dla omawianego systemu laboratoryjnego wykorzystano środowisko **Quartus**, które jest jednym z bardziej przyjaznych narzędzi **CAD** (Computer Aided Design).

- W jego skład wchodzi edytor graficzny, edytor tekstowy HDL, kompilator, symulator funkcjonalny i czasowy, bogate biblioteki gotowych bloków, system definiowania stylów kompilacji projektu.

- Pakiet ten umożliwia projektowanie w obu standardowych językach **HDL**, tj. w języku **VHDL** oraz **Verilog**.

- Pakiet umożliwia wprowadzenie i edycję projektu, kompilację, określenie docelowego układu programowalnego, przyporządkowanie wyprowadzeń, symulację czasową oraz zaprogramowanie układu.

- Do budowy omawianego stanowiska laboratoryjnego wykorzystano **edytor tekstowy** języka opisu sprzętu VHDL.

Podsumowanie - walory dydaktyczne projektu

- Projekt ma charakter wieloetapowy i rozwojowy.
- Powinien być realizowany przez studentów ostatnich semestrów.
- Stanowi przykład sytemu Internetu rzeczy.
- Student przystępujący do projektu musi posiadać wiedzę z następujących dziedzin:
  - **µk**– architektura i programowanie w języku C,
  - komunikacja poprzez żądania http w sieci Ethernet,
  - tworzenie stron internetowych w języku HTML z podstawowymi komponentami formularza,
  - protokół komunikacyjny magistrali SPI,
  - architektura i programowanie układów **FPGA** w języku VHDL.

- Student realizujący projekt rozwija umiejętności przez wykorzystanie:
  - **interfejsu AJAX** do modyfikacji stanu wybranych komponentów strony internetowej,
  - **metod obiektu** XMLHttpRequest do usprawnienia wymiany danych między klientem i serwerem,
  - procesów równoległych wykonywanych przez struktury FPGA.

- Student realizujący projekt poznaje i oprogramowuje wiele urządzeń cyfrowych stosowanych powszechnie w systemach Internetu rzeczy, np. wyświetlacz LCD, potencjometr cyfrowy, termometr cyfrowy, pamięć EEPROM.
- Projekt można prowadzić przez cały semestr, a jego zakres dostosować do indywidualnych pomysłów studenta i prowadzącego.
- Studenci chętnie podejmują działania zapewniające im samodzielność, jednak z możliwością i wymogiem konsultacji z prowadzącym.
- Ukończenie projektu umacnia zawodową pewność siebie studenta.

- W opracowanym i zbudowanym stanowisku aspekt e-learningu polega na wykorzystaniu środowiska programistycznego do zdalnego ładowania programu µk przez sieć Ethernet.
- Do pamięci µk można załadować po Ethernecie nie tylko program wykonywalny, ale również stronę internetową użytkownika systemu napisaną w języku HTML (jeśli nie przekracza ona rozmiarów pamięci programu µk).
- Prezentowany projekt bazowy ma **charakter rozwojowy**, student otrzymuje podstawowe procedury biblioteczne dla wybranych urządzeń sytemu z interfejsem SPI i ma za zadanie wzbogacić architekturę o kolejne urządzenia oraz bibliotekę o kolejne procedury.

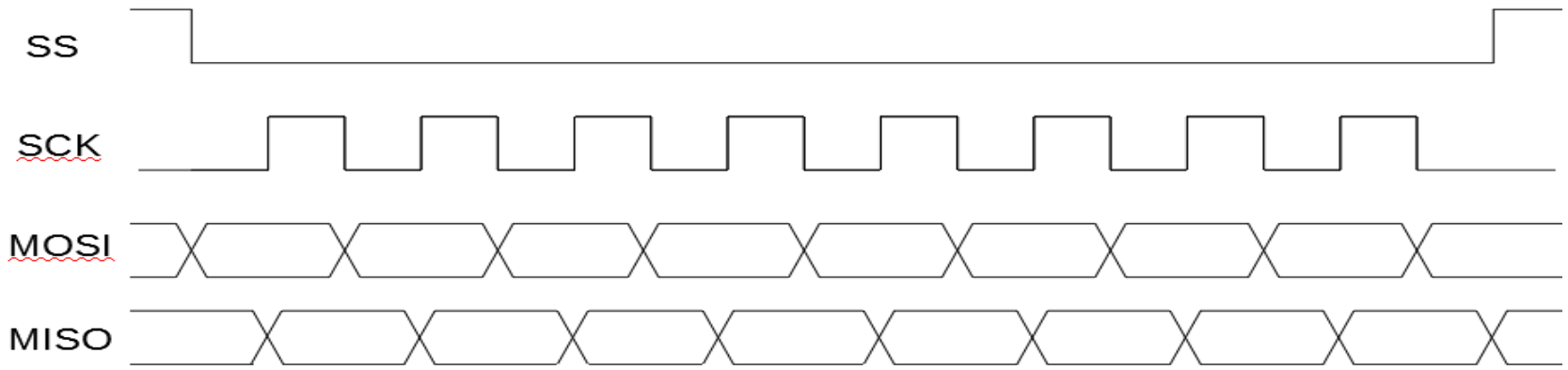
Dziękuję za uwagę

CHARAKTERYSTYKA MAGISTRALI SPI cd.

➢Podczas każdego taktu zegara magistrali SPI zachodzi transmisja dupleksowa:
- **Master** nadaje jeden bit na linii MOSI,
- **Slave** odczytuje ten bit i w tym samym takcie zegara SPI, nadaje jeden bit po linii MISO odczytywany przez Master.

➢Urządzenia Master i Slave działają w tej transmisji jak rejestry przesuwne – w ośmiu taktach zegara SPI bajt danych początkowo znajdujący się w rejestrze danych Master zostanie przesłany do Slave, a bajt z rejestru danych urządzenia Slave znajdzie się w Master.

➢ Dwie linie danych magistrali MOSI i MISO – każda dla jednego kierunku transmisji, tworzą magistralę dookólną, na której w każdym takcie zegara dochodzi do nadania jednego bitu i odebrania jednego bitu.



Przebiegi na magistrali SPI między mikrokontrolerem a układem CPLD