**Organizacja i przeprowadzenie specjalistycznych szkoleń podnoszących kompetencje dla pracowników Wydziału Informatyki i Wydziału Mechatroniki oraz szkoleń zawodowych dla studentów kierunku Informatyka i mechatronika Uniwersytetu Kazimierza Wielkiego w Bydgoszczy w ramach projektu „Stawiamy na rozwój UKW”**

**CZĘŚĆ 1 – SZKOLENIE DLA PRACOWNIKÓW Z JĘZYKA PYTHON**

1. **Podstawy programowania w języku Python - szkolenie dla nie programistów** - min. 5 dni (40 godzin).

Szkolenie stacjonarne u zamawiającego dla min. 6 osób, zamawiający udostępnia nieodpłatnie wykonawcy salę dydaktyczną wyposażoną w ławki, projektor oraz dostęp do internetu.

Wykonawca zapewnia materiały szkoleniowe, certyfikat uczestnictwa w szkoleniu, laptopy szkoleniowe dla uczestników.

Szkolenie realizowane metodami: mikrowykład, ćwiczenia wspólne, ćwiczenia indywidualne, praca indywidualna. Duży nacisk należy położyć na aspekt praktyczny programowania.

Minimalny zakres tematyczny szkolenia:

Język Python

* Geneza
* Zastosowania
* Mocne i słabe strony języka
* Środowisko kompilacyjne, wersjonowanie

IDE (Zintegrowane Środowisko Programistyczne) dla Pythona

* Porównanie różnych IDE dla języka Python
* PyCharm a Microsoft Visual Studio 2019
* Instalacja i konfiguracja środowisk
* Strona www.python.org i jej zawartość

Wstęp do programowania

* Co to są języki programowania?
* Algorytm – definicja oraz użycie
* Instrukcje programistyczne
* Przykłady programowania wybranych algorytmów. Jak algorytmy prezentują się w kodzie źródłowym?
* Co to są zmienne i typy danych?
* Analiza wybranych struktur języków programowania

Typy danych w języku Python

* Rodzaje typów wbudowanych (wewnętrznych) – operacje na danych oraz metody specyficzne dla typu
* Liczby
* Łańcuchy znaków (tekst)
* Logiczne typy danych (prawda, fałsz)
* Struktury danych – ich charakterystyka i metody
* Listy
* Krotki
* Zbiory
* Słowniki

Instrukcje Sterujące (operacje I/O, przypisania, warunki i pętle)

* Podstawowe zasady tworzenie kodu w języku Python – znaczenie wcięć. Jak pisać w Pythonie?
* Dobre praktyki w pisaniu kodu – PEP8 – dokument definiujące reguły gramatyczne języka Python
* Idea instrukcji sterujących
* Przypisanie
* Operacje wejścia/wyjścia
* Instrukcje warunkowe (if)
* Instrukcje iteracyjne – pętle (while i for)
* Wielokrotne powtarzanie czynności – operacje na typach w strukturach danych.

Zasady tworzenia dokumentacji w języku Python

* Komentarze
* Notatki dokuemntacyjne (\_\_doc\_\_)
* PyDoc

Pliki

* Pliki binarne i tekstowe
* Ścieżki i dostęp do plików
* Operacje na plikach
* Pliki CSV

Funkcje

* Podstawy funkcji
* Zasady tworzenia i nazewnictwa
* Wywoływanie funkcji
* Polimorfizm w funkcjach
* Parametry w funkcjach
* Wybrane aspekty użycia funkcji
* Funkcje zagnieżdżone
* Zasięgi
* Argumenty funkcji

Zaawansowane zagadnienia dotyczące funkcji

* Fukcje rekurencyjne
* Funkcje Lambda
* Listy składane i generatory

Wstęp do tworzenie modułów

* Tworzenie modułów
* Używanie modułów
* Pakiety modułów
* Przestrzenie nazw

Wstęp do programowania obiektowego w języku Python

* Co to jest programowanie obiektowe?
* Definicja obiektu
* Paradygmaty programowania obiektowego
* Obiekty
* Czym jest klasa?
* Tworzenie klas – instancja, konstruktor, metody
* Dziedziczenie – wielodziedziczenie
* Przeciążanie operatorów – tworzenie różnych wariantów operatorów
* Klasy mieszane
* Klasy abstrakcyjne, interfejsy

Wyjątki, obsługa błędów

* Podstawy obsługi wyjątków – hierarchii
* Instrukcja try/except/else/finally
* Instrukcja raise
* Instrukcja assert
* Klasy wyjątków

Przetwarzanie danych XML w języku Python

* Wprowadzenie do języka XML
* Parsery DOM i SAX
* Biblioteka xml.etree.ElementTree
* Document Object Model API – mindom (implementacja minimal)
* Parsery SAX2
* SAX handler
* Użycie expat do szybkiego parsowania XML

Wstęp do programowania wielowątkowego

* Moduły thread, threading oraz concurrent.futures
* Synchronizacja wątków
* Równoległe wykonywanie funkcji z różnymi argumentami
1. **Podstawy programowania w języku Python - szkolenie dla programistów** - min. 4 dni (32 godziny).

Szkolenie stacjonarne u zamawiającego dla min. 6 osób, zamawiający udostępnia nieodpłatnie wykonawcy salę dydaktyczną wyposażoną w ławki, projektor oraz dostęp do internetu.

Wykonawca zapewnia materiały szkoleniowe, certyfikat uczestnictwa w szkoleniu, laptopy szkoleniowe dla uczestników.

Szkolenie realizowane metodami: mikrowykład, ćwiczenia współne, ćwiczenia indywidualne, praca indywidualna. Duży nacisk należy położyć jest na aspekt praktyczny programowania.

Minimalny zakres tematyczny szkolenia:

Język python

* Geneza
* Zastosowania
* Mocne i słabe strony języka
* Środowisko kompilacyjne, wersjonowanie
* DE dla Pythona
* Porównanie różnych IDE dla języka Python
* PyCharm a Microsoft Visual Studio 2019
* Instalacja i konfiguracja środowisk
* Strona www.python.org i jej zawartość

Typy danych w języku Python

* Rodzaje typów wbudowanych – opracje na danych oraz metody specyficzne dla typu
* Liczby
* Łańcuchy znaków
* Logiczne typy danych

Struktury danych – ich charakterystyka i metody

* Listy
* Krotki
* Zbiory
* Słowniki

Instrukcje Sterujące

* Podstawowe zasady tworzenie kodu w języku Python – znaczenie wcięć
* Dobre praktyki w pisaniu kodu – PEP8 – dokument definiujące reguły gramatyczne języka Python
* Przypisanie
* Operacje wejścia/wyjścia
* Instrukcje warunkowe (if)
* Instrukcje iteracyjne – pętle (while i for)
* Iteracja – operacje na typach strukturach danych

Zasady tworzenia dokumentacji w języku Python

* Komentarze
* Notatki dokumentacyjne (\_\_doc\_\_)
* PyDoc

Pliki

* Pliki binarne i tekstowe
* Ścieżki i dostęp do plików
* Operacje na plikach
* Pliki CSV

Funkcje

* Podstawy funkcji
* Zasady tworzenia i nazewnictwa
* Wywoływanie funkcji
* Polimorfizm w funkcjach
* Parametry w funkcjach

Wybrane aspekty użycia funkcji

* Funkcje zagnieżdżone
* Zasięgi
* Argumenty funkcji

Zaawansowane zagadnienia dotyczące funkcji

* Funkcje rekurencyjne
* Funkcje Lambda
* Listy składane i generatory

Wstęp do tworzenie modułów

* Tworzenie modułów
* Używanie modułów
* Pakiety modułów
* Przestrzenie nazw

Wstęp do programowania obiektowego w języku Python

* Paradygmaty programowania obiektowego
* Obiekty
* Tworzenie klas – instancja, konstruktor, metody
* Dziedziczenie – wielodziedziczenie
* Przeciążanie operatorów
* Klasy mieszane
* Klasy abstrakcyjne, interfejsy

Wyjątki, obsługa błędów

* Podstawy obsługi wyjątków – hierarchia
* Instrukcja try/except/else/finally
* Instrukcja raise
* Instrukcja assert
* Klasy wyjątków

Przetwarzanie danych XML w języku Python

* Wprowadzenie do języka XML
* Parsery DOM i SAX
* Biblioteka xml.etree.ElementTree
* Document Object Model API – mindom (implementacja minimal)
* Parsery SAX2
* SAX handler
* Użycie expat do szybkiego parsowania XML

Wstęp do programowania wielowątkowego

* Moduły thread, threading oraz concurrent.futures
* Synchronizacja wątków
* Równoległe wykonywanie funkcji z różnymi argumentami
1. **Warsztat - programowanie w języku Python - poziom średniozaawansowany** - min. 4 dni (32 godziny).

Szkolenie stacjonarne u zamawiającego dla min. 4 osób, zamawiający udostępnia nieodpłatnie wykonawcy salę dydaktyczną wyposażoną w ławki, projektor oraz dostęp do internetu.

Wykonawca zapewnia materiały szkoleniowe, certyfikat uczestnictwa w szkoleniu, laptopy szkoleniowe dla uczestników.

Szkolenie prowadzone jest jako warsztat – laboratorium. W trakcie warsztatu uczestnik pracuje indywidualnie i grupowo pod okiem doświadczonego trenera.

Minimalny zakres tematyczny szkolenia:

Analiza Środowiska programistycznego dla Pythona

* Konfiguracja środowiska
* Podstawowe zasady korzystania ze środowiska
* Proces debuggingu
* Analiza błędów

Zasady programowania w języku Python

* Styl PEP8
* Interpolacja ciągów string
* Wdrożenie zasad programowania i konstruowania programów
* Używanie funkcji i obiektów
* Zasady użycia instrukcji

Programowanie kolekcji

* Kolekcje: listy, krotki, zbiory oraz słowniki
* Operacje na danych
* Stosowanie dedykowanych metod dla kolekcji
* Asocjacje key – value
* Zastosowania metody \_\_missing\_\_()

Podstawy programowania funkcyjnego

* Pisanie prostych funkcji
* Paradygmaty programowania funkcyjnego
* Funkcje rekurencyjne
* Iteratory i generatory
* Użycie funkcji do przetwarzania danych: format XML
* Wprowadzenie elementów dynamicznych do programowania funkcyjnego
* Funkcje wyższego rzędu: lambda, sorted(), filter(), iter(), map() i inne
* Funkcje zwracające n-wyników
* Definiowanie zachowań opcjonalnych
* Metoda yield() – użycie wielu generatorów

Własny projekt uczestnika szkolenia – programowanie funkcyjne

* Wykonanie indywidualnego projektu mini-aplikacji opartej na modelu funkcyjnym programowania
* Napisanie kodu źródłowego prostej aplikacji opartej na programowaniu funkcyjnym
* Analiza błędu i testy własnej aplikacji

Programowanie obiektowe

* Pisanie obiektów
* Diagram klas, jak go czytać?
* Programowanie klas
* Paradygmaty programowania obiektowego: Polimorfizm, Enkapsulacja, Dziedziczenie, Abstrakcja
* Porównanie rozwiązań obiektowych i funkcyjnych
* Klasy abstrakcyjne i interfejsy
* Praktyczne aspekty programowania obiektowego

Własny projekt uczestnika szkolenia – programowanie obiektowe

* Wykonanie indywidualnego projektu mini-aplikacji opartej na modelu obiektowym programowania
* Napisanie kodu źródłowego prostej aplikacji opartej na programowaniu obiektowym
* Analiza błędu i testy własnej aplikacji

Programowanie i modelowanie struktur danych

* Pliki tekstowe
* Pliki XML
* Pliki JSON
* Proces parsowania danych
* Sprawdzanie poprawności danych

Obsługa wyjątków i wydajność

* Procesy testowanie i debugowania
* Konstrukcja try/except/else/finally
* Analiza wyników testów
* Optymalizacja programu
* Analiza wykorzystania pamięci, wycieki pamięci

Własny projekt uczestnika szkolenia – programowanie parsera źródła danych

* Wykonanie indywidualnego projektu mini-aplikacji parsującej wybrany model źródła danych
* Napisanie kodu źródłowego zaprojektowanej aplikacji
* Analiza błędu i testy własnej aplikacji

Cykl życia aplikacji języka Python

Analiza zastosowań języka Python

1. **programowanie w języku Python - poziom zaawansowany** - min. 4 dni (32 godziny).

Szkolenie stacjonarne u zamawiającego dla min. 6 osób, zamawiający udostępnia nieodpłatnie wykonawcy salę dydaktyczną wyposażoną w ławki, projektor oraz dostęp do internetu.

**Wykonawca zapewnia materiały szkoleniowe, certyfikat uczestnictwa w szkoleniu, laptopy szkoleniowe dla uczestników**

**Szkolenie jest realizowane metodami: mikrowykład, ćwiczenia wspólne, ćwiczenia indywidualne, praca indywidualna, z głębokim naciskiem położonym na aspekt praktyczny programowania.**

Minimalny zakres tematyczny szkolenia:

Środowisko pracy: omówienie

Interpreter języka Python

* Style PEP8
* IDE dla Pythona

Zaawansowane aspekty typowania

* Typowanie dynamiczne
* Łańcuchy znaków
* Generalne kategorie typów
* Typy złożone

Programowanie funkcyjne

* Pisanie funkcji prostych i złożonych
* Paradygmaty programowania funkcyjnego
* Funkcje iteracyjne, funkcje: zip(), reverse(), enumerate()
* Funkcje wyższego rzędu, funkcje generatorowe
* Złożone funkcje rekurencyjne
* Redukcje
* Moduły: itertools, functools, biblioteka PyMonad
* Dekoratory
* Złożone funkcje Lambda
* Funkcyjne programowanie wielowątkowe

Programowanie obiektowe

* Klasy, Interfejsy i metody
* Paradygmaty programowania obiektowego
* Klasy abstrakcyjne
* Kontenery niestandardowe
* Przeciążanie operatorów
* Programowanie zorientowane obiektowo
* Tworzenie klas mieszanych
* Rozszerzanie typów wbudowanych
* Dziedziczenie diamentowe – zmiany w wersji 3.x Pythona
* Klasy w nowym stylu
* Metody statyczne i metody klasy
* Pułapki związane z klasami

Moduły

* Przestrzenie nazw modułów
* Pakiety modułów
* Ukrywanie danych w modułach
* Przechodnie przeładowywanie modułów
* Pułapki związane z modułami

Metaprogramowanie

* Metaklasy – model metaklasy
* Tworzenie i dekorowanie metaklas
* Algorytm dziedziczenie w Pythonie
* Atrybuty, Atrybuty getter i setter
* Refaktoryzacja atrybutów a @property
* Podklasy i klasy nadrzędne
* Zastosowanie metaklas

Współbieżność i równoległość

* Tworzenie wątków
* Programowanie równoległe
* Klasy współbieżne: Lock, Queue, ThreadPoolExecutor
* Asynchroniczność
* Pułapki wiązane z programowaniem współbieżnym i asynchronicznym

Wydajność i optymalizacja

* Testowanie aplikacji w języku Python
* Analiza błędów
* Profilowanie
* Zarządzanie pamięcią
* Testowanie wydajności

Zaawansowane aspekty programowania w języku Python

* Wzorce projektowe
* Systemy rozproszone
* Przetwarzanie języka naturalnego (NLP)

Bazy danych

* Połączenie z bazą danych SQL
* Systemy ORM
* Interpreter sqlite Pythona, baza SQLITE3
* Peewee
* Bazy noSQL

**CZĘŚĆ 2 – SZKOLENIE LOTNICZE WRAZ Z OPŁATĄ WYDANIA LICENCJI PILOTA, ZGODNIE Z WYTYCZNYMI ULC I EASA DLA PRACOWNIKÓW ORAZ STUDENTÓW**

**VLOS - NSTS-01**

Szkolenie na operatora bezzałogowych statków powietrznych z uprawnieniami VLOS (uprawnienie do lotów w zasięgu wzroku operatora oraz przy użyciu danych telemetrycznych) w ramach scenariusza NSTS-01.

Szkolenie dla min. 10 studentów oraz 4 pracowników dydaktycznych.

Szkolenie zawiera takie elementy jak:

* zajęcia teoretyczne min. 12 godzin zegarowych nie wliczając w to przerw (wykłady), przeprowadzone metodą stacjonarną w siedzibie zamawiającego,
* praktyka lotnicza na ziemi – min. 1 godzina zegarowa/osobę
* praktyka lotnicza na lądowisku – min. 3 godziny zegarowe /osobę
* egzamin wewnętrzny z omówieniem wyników;
* egzamin Państwowy

Szkolenie zgodnie z programem nauczania EASA który kończy się **egzaminem Państwowym,** którego celem jest zdobycie uprawnienia podstawowego - NSTS-01, po którym kursant staje się członkiem personelu lotniczego UE oraz otrzyma Licencję Pilota UAVO.

Szkolenie w tym scenariuszu będzie trwało min. 3 dni.

Cena zawiera szkolenie do uprawnienia NSTS-01 (VLOS do 4 kg) oraz opłatę dot. wydania licencji pilota.

Szkolenie realizowane w siedzibie zamawiającego (zamawiający udostępni salę wyposażoną w ławki, i projektor).

Czas realizacji: lipiec - październik 2023

**BVLOS - NSTS-05**

Szkolenie na operatora bezzałogowych statków powietrznych z uprawnieniami BVLOS (uprawnienie do lotów poza zasięgiem wzroku operatora oraz przy użyciu danych telemetrycznych) w ramach scenariusza NSTS-05.

Szkolenie dla min. 2 pracowników dydaktycznych.

Szkolenie zawiera takie elementy jak:

* zajęcia teoretyczne min. 16 godzin zegarowych nie wliczając w to przerw (wykłady), przeprowadzone metodą stacjonarną w siedzibie zamawiającego,
* praktyka lotnicza na ziemi – min. 1 godzina zegarowa/osobę
* praktyka lotnicza na lądowisku – min. 8 godziny zegarowe /osobę
* egzamin wewnętrzny z omówieniem wyników;
* egzamin Państwowy

Szkolenie zgodnie z programem nauczania EASA który kończy się egzaminem Państwowym, którego celem jest zdobycie uprawnienia podstawowego - NSTS-05, po którym kursant staje się członkiem personelu lotniczego UE oraz otrzyma Licencję Pilota UAVO.

Szkolenie w tym scenariuszu trwa min. 4 dni.

Cena zawiera szkolenie do uprawnienia NSTS-05 (BVLOS do 4 kg) oraz opłatę dot. wydania licencji pilota.

Cena zawiera wszystkie koszty związane z realizacją szkolenia.

Szkolenie realizowane w siedzibie zamawiającego (zamawiający udostępni salę wyposażoną w ławki, i projektor).

**Czas realizacji: lipiec - październik 2023**

**Instruktorzy z wyższym wykształceniem, minimum jeden instruktor z przygotowaniem pedagogicznym, praca w systemie instruktor-uczeń.**

**CZĘŚĆ 3 – SZKOLENIE DLA SUDENTÓW „INTERNET RZECZY”**

Liczebność grupy 12-13 osób, 8 grup = 100 osób

Szkolenie dla 100 osób, w łącznym wymiarze 194 godz. dydaktycznych zajęć, realizowane będzie dla 8 grupach studentów (po ok. 12-13 osób w grupie), w wymiarze 24 godzin dydaktycznych zajęć dla jednego uczestnika/1grupa;

Uczestnicy szkoleń: studenci kierunku mechatronika i informatyka prowadzonych przez UKW Bydgoszcz.

Zamawiający na potrzeby realizacji szkolenia (dla każdej grupy) nieodpłatnie udostępni sale szkoleniowe wyposażone w: biurko prowadzącego, ławki słuchaczy kursu, rzutnik multimedialny i ekran, komputer umożliwiający prezentację materiałów dydaktycznych.

Pozostałe elementy niezbędne do przeprowadzenia szkolenia zapewnia Wykonawca.

Wykonawca realizuje proces pomiaru kompetencji przed i po szkoleniu oraz w ramach szkolenia realizuje proces certyfikacyjny i wydaje certyfikaty.

Zamawiający wymaga, aby w ramach szkolenia dla każdej grupy, zostały zrealizowane co najmniej następujące zagadnienia merytoryczne (program szkolenia):

**Moduł 1 - OpenSource’owy system automatyki domowej - 8h**

 Zakres merytoryczny:

* Systemy opensource,
* Protokoły w automatyce domowej,
* Platforma sprzętowa,
* Instalacja i podstawowa konfiguracja systemu Raspbian
* Instalacja usługi Domoticz lub OpenHub
* Tworzenie użytkowników i konfiguracja usługi,
* Zdalny dostęp do systemu,
* Instalowanie urządzeń IoT w systemie,
* Tworzenie skryptów automatyki,
* Tworzenie planu pomieszczeń z naniesionymi urządzeniami
* Aplikacja mobilna do zarządzania systemem,
* Nakładki systemowe innych dostawców i ich instalacja w systemie
* Monitorowanie pracy systemu automatyki

Niezbędne wyposażenie:

* Komputer jednopłytkowy z otwartym systemem operacyjnym wyposażony w min.: 1xHDMI, H.246, MPG 4 decode, H. 246 encode, OperGL ES 2.0, Quad-Core ARM Cortex-A53 1,4 GHz, 1 GB LPDDR2 @ 900 MHz, Ethernet 300 Mbps, 2,4 GHz i 5 GHz, 802.11 b/g/n/ac, BLE 4.2 - sztuk 10;
* karty micro SD 8 GB co najmniej class 10 - 10 sztuk;
* klawiatury usb i myszki/ lub klawiatura z touchpadem - 10 sztuk
* moduły z-wave - 10 sztuk
* przekaźniki wifi (np. Blebox, Shelly) - 10 sztuk
* czujniki ruchu pir - 10 sztuk
* czujniki temperatury - 10 sztuk
* makieta domu z czujnikami ruchu, temperatury, przekaźnikami wifi - 5 sztuk.

**Moduł 2 - Rozwiązania techniczne dla Internetu Rzeczy - Montaż elementów elektronicznych - 8h**

Zakres merytoryczny:

Przygotowanie układu sprzętowego do obsługi programistycznej. Lutowanie układów elektronicznych z wykorzystaniem płytek prototypowych (użycie typowych elementów przewlekanych takich jak diody, rezystory itp.) oraz gotowego zestawu do samodzielnego montażu: przetwornik A/C do PC z komunikacją przez port USB, z możliwością obsługi programistycznej przez port wirtualny com z poziomu języka C# w środowisku Windows. Układ przetwornika ma zostać złożony i przetestowany przez studentów w trakcie szkolenia.

Niezbędne wyposażenie - zestaw do nauki lutowania:

* stacje lutownicze (minimalne wymagania co do specyfikacji): układ regulacji temperatury z czujnikiem w grzałce umożliwiający regulację w zakresie 150°C + 480°C, uziemienie grota, podstawka z czyścikiem grotów, grot lutowniczy stożkowy 0,8mm, po jednej dla każdego uczestnika szkolenia,
* niezbędne materiały do lutowania,
* płytki prototypowe,
* elementy składowe i płytka dla przetwornika AC/PC (zasilanie 5V, połączenie z komputerem poprzez port RS232 lub USB), niezbędnego do realizacji zadań z zakresu lutowania przewidzianych programem szkolenia;

**Moduł 3 - Interfejsy komunikacyjne w sieciach IOT/IIOT oraz systemach smart-home - 8h**

Zakres merytoryczny:

* omówienie trybów pracy oraz sposobów programowania modułów ESP8266 i ESP32
* wykorzystanie gotowego firmware’u z interfejsem graficznym (typu ESP Easy) do budowy własnego systemu IOT
* przesyłanie danych na platformy IOT typu ThingSpeak
* budowa i uruchomienie własnego modułu IOT

Niezbędne wyposażenie:

* co najmniej 1 punkt dostępowy sieci WiFi z możliwością komunikacji przez uczestników zajęć
* moduł rozwojowy WiFi oparty na układzie ESP8266-12F, 10-bitowe ADC, napięcie 5V, 3.3V, 10GPIO (WPM, I2C), min. 4 MB Flash, microUSB, USB-UART CP2102, PCB, IEEE 802.11 b/g/n (Station/SoftAP/SoftAP + Station), programowane za pomocą języka Lua, MicroPython, komend AT, Arduino, itp., - w ilości przynajmniej 1 szt. dla każdego uczestnika zajęć
* moduł rozwojowy wyposażony w układ komunikacji bezprzewodowej ESP32, GPIO, I2C, I2S, SPI, UART, 2,4 GHz, WiFi 802.11 b/g/n, zasilanie od 3,3 V do 5 V, flash: 4 MB, PSRAM: 2 MB, antena wbudowana PCB, microUSB, zasilanie: microUSB , 5V, 3.3V, przyciski: Boot, Reset - w ilości przynajmniej 1 szt. dla każdego uczestnika zajęć
* co najmniej 2 czujniki dla każdego uczestnika zajęć, które można podłączyć do obu modułów rozwojowych
* kompletny zestaw okablowania, który umożliwia programowanie obu modułów rozwojowych oraz podłączenie czujników wykorzystywanych w trakcie zajęć
* układ zasilania do modułów rozwojowych pozwalający na przeprowadzenie zajęć
* programowalna platforma mobilna 2 szt., wyposażona w komunikację USB, bluetooth, wifi (tryb acces point i karty sieciowej), wyposażona minimum w dwa koła skrętne sterowane za pomocą serwomechanizmu, napęd z wykorzystaniem silnika DC, wyposażona w wymienne czujniki: odległości 2 szt., akcelerometr 1 szt., możliwość programowanie w języku c++, wyświetlacz OLED.

Prowadzący szkolenia musi umożliwić udział w szkoleniu pracownikowi UKW w przypadku konieczności weryfikacji realizowanego materiału.

Przy składaniu oferty należy załączyć listę sprzętu, którego oferent zamierza użyć do realizacji szkoleń.