|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | | | **Protokół konstrukcyjny układu wykonawczego-software** | | | | | | Data wystawienia: | |
|  | 27/12/2021 | |
|  | Doc# | 1/CZB/006 |
|  | Nr wniosku NCBR: | | POIR.01.01.01-00-0196/19 | | | Nazwa projektu: | | Smart Yacht |
|  | Rozpoczęcie testów: | | 01-12-2021 | | Zakończenie testów: | | 27-12-2021 | |

#### Cel programu

Celem jest napisanie software obsługującego inicjator(opaskę na rękę) oraz responder(układ stacjonarny na łodzi), oraz wyznaczającego dystans metodą ToF kóry następnie zostanie wysłany magistralą CAN do mastera.

Docelowo ma on obsługiwać 6 responderów wymaganych do wykonania trilateracji 6 punktowej, oraz 9 inicjatorów do śledzenia 9 członków załogi.

#### **Program respondera**

Kod jest bazowany na przykładzie dystansowania UWB.

Na początku zostaje załączone API czujników UWB.

W strukturze typu dwt\_config\_t o nazwie config zostają ustawione parametry transmisji danych.

Tablice rx\_poll\_msg[], tx\_resp\_msg[] oraz rx\_final\_msg[] zawierają wzory ramek, zmienna responderNumber przechowuje wstępnie zaprogramowany numer reposndera, a actualInitiatorNumber przechowuje numer aktualnie dystansowanego inicjatora.

W pętli głównej responder oczekuje na sygnał odebrany od inicjatora. Jeżeli taki zostaje odebrany, z ramki zostaje odczytany oraz przypisany do actualInitiatorNumber numer inicjatora. Po zweryfikowaniu poprawności ramki zostaje zapisany timestamp, który jest następnie odsyłany do inicjatora. Reponder oczekuje na kolejną ramkę. Po jej odebraniu wylicza dystans na podstawie różnicy pomiędzy zapisanym poprzednim TS a aktualnie odebranym od inicjatora. Wyliczony dystans zostaje przesłany przez CAN, podając numer respondera(zawięrający się w adresie CAN) oraz 5 bajtową ramkę zawierającą numer zdystansowanego inicjatora, 2 bajty zawierające dystans w centrymetrach. Oraz stan baterii i informacją o stanie opaski(podłaczony/ładowanie/rozładowywanie).

#### Program inicjatora

Konfiguracja parametrów transmisji oraz wzory ramek zostały ustalone podobnie jak w responderze.

Program został skonstruowany w taki sposób, żeby inicjator usypiał się na około 1s a następnie wybudzał się i wykonywał procedurę skanowania po kolei wszystkich responderów.

Jest to działanie niezbędne w celu utrzymania niskiego poboru prądu układu- będzie on zasilany z niewielkiego akumulatora litowo-jonowego.

Dokładny czas spania układu różni się w zalezności od numeru inicjatora- działanie to ma na celu rozsynchronizowywanie się poszczególnych inicjatorów pomiędzy sobą, ponieważ transmisja nie może pracować jednocześnie na tym samym kanale pomiędzy dowolnymi urządzeniami w zasiągu. W przypadku kiedy mimo to nastąpi jednoczesna komunikacja więcej niż jednej pary inicjator-responder jedna z nich nie dojdzie do skutku. O wybudzania MCU dba wbudowany weń zegar czasu rzeczywistego. W przypadku gdy opaska jest ładowana bądź podłaczana, opaska jest wybudzana co kilkadziesiąt sekund- dzięki temu nie przeszkadza w komunikacji aktywnych opasek, oraz szybciej się ładuje. W momencie położenia opaski na ładowarce, w ostatniej wysłanej w krótkim interwale ramce zawiera informację o tym, że została podłaczona do ładowania.

#### Wnioski

Układ został przetestowany przy użyciu 2 responderów oraz 3 inicjatorów. Umożliwia każdy z responderów jest w stanie w ciągu minimum 5 sekund zdystansować wszystkie 3 inicjatory.

Wykonał:

Bartosz Pracz