

Versuch 5: CAN

Übertragung von Messdaten zwischen Nachbararbeitsplätzen

Die Mikrocontrollerarbeitsplätze einer Labortischreihe sind jeweils mittels einer CAN-Schnittstelle miteinander verbunden. Die CAN-Schnittstelle arbeitet mit den durch DAVE defaultmäßig eingestellten 100kBit/sec, auch das Bittiming bleibt entsprechend der Voreinstellung. Es wird ausschließlich mit den Standard-Identifiern (11-Bit) gearbeitet.

Folgende Messdaten werden durch die unterschiedlichen Arbeitsplätze angeboten bzw. erwartet:

Arbeitsplatz	zu sendende Messgröße	Identifizier	zu empfangende Messgröße	Identifizier
Fensterseite	Temperatur	0x1nn	Gewicht	0x2mm
Mitte	Gewicht	0x2nn	Motordrehzahl	0x3mm
Türseite	Motordrehzahl	0x3nn	Temperatur	0x1mm

nn ist die zweistellige Nummer des eigenen PC, mm die des fremden PC's.

Jeder Arbeitsplatz hat zunächst ein TRANSMIT OBJECT und ein RECEIVE OBJECT gemäß obiger Tabelle einzurichten. Das Empfangen soll interruptgesteuert erfolgen, die Interruptquelle ist RXIE des "message control registers".

Für die Behandlung des Datenarrays der Objekte empfiehlt sich die Einführung der folgenden Union:

```
union typ_uMy{
    char cArFrm[8];           // Darstellung als Array
    float fFrm;               // Darstellung als float-Wert
};
typedef union typ_uMy t_uMyDates;
```

Gemeinsam mit dem durch DAVE eingeführten Strukturtyp TCAN_Obj kann das Empfangen von Daten mittels der beiden folgenden Variablen durchgeführt werden:

```
TCAN_Obj sRcvMyObj;         // Übernimmt das Objekt in der ISR
t_uMyDates uMyRcvDates;     // Übernimmt die Daten aus sRcvMyObj
                             // als Array und ermöglicht den Zugriff
                             // als float-Wert
```

Dadurch ist für die richtige Interpretation der Daten der Sende- und Empfangsseite Sorge getragen.

Für das TRANSMIT OBJECT genügt es lediglich die folgende Variable zur Verfügung zu stellen und geeignet zu versorgen:

```
t_uMyDates uMyTrmDates;
```

Da alle weiteren Eigenschaften des Objektes unverändert bleiben, genügt es mittels der DAVE-Funktion CAN_vLoadData() das Objekt mit neuen Daten zu aktualisieren.

a) Realisieren Sie die beiden CAN-Objekte mittels der notwendigen Einstellungen in DAVE!

Die Funktion mit dem Prototypen

```
bit bCanRok(void)
```

wird in der Hauptschleife von main() aufgerufen und liefert "TRUE", sobald eine CAN-Message empfangen wurde. Sie gibt den Wert eines in der ISR nach erfolgreichem Empfang des RECEIVE OBJECTS gesetzten Flags (bCanFlag) zurück!

Die folgende Funktion, wird im Erfolgsfall ausgeführt und liefert den eingetroffenen Wert zurück:

```
float fGiveCanValue(void);
```

Dieser Wert wurde vor dem Setzen von bCanFlag in der ISR in die modulglobale Variable uMyRcvDates kopiert!

- b) Realisieren sie die beiden Funktionen als Teil des CAN-Quellcode-Moduls, die Anzeige des übernommenen Wertes geschieht in main(). Dazu wird in der 1. Zeile des Displays durch eine 2-stellige Zahl die Gesamtzahl der bisher eingetroffenen CAN-Nachrichten angegeben, durch einen Doppelpunkt getrennt ist dahinter der empfangene Messwert auszugeben!
- c) Durch Drücken der Taste 2 ist das TRANSMIT OBJECT aufzubereiten und zu versenden. Realisieren Sie zu diesem Zweck die Funktion

```
void vMakeCanValue(float);
```

die den übergebenen Wert an das Sendeobjekt weiterreicht und die Übertragung veranlasst! Vergewissern Sie sich anhand Simulation über die Richtigkeit Ihres Programmes! Testen Sie gemeinsam mit Ihren Nachbarn im Target!

optional:

- d) Erweitern Sie Ihr Programm um die Möglichkeit mittels "Remote-Frames" Werte direkt anzufordern!

-- Bitte gehen Sie rücksichtsvoll und sorgsam mit Ihrer Nachbargruppe um !! --