1. В функции apply\_can\_data (синим что надо добавить):

#if defined(FOR\_TRUCK)

////////////////////////////////////////////////////////////////////////////////

// Fuel Level - Уровень топлива

if (FUELLEVELbits.can\_number != 0) {

if (FUELLEVELbits.can\_number == 1 && (FUELLEVELbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[0] == 0xFF) && (CAN1\_DATA\_buf[2] == 0xFF) && (CAN1\_DATA\_buf[3] == 0xFF) && (CAN1\_DATA\_buf[4] == 0xFF) &&

// (CAN1\_DATA\_buf[5] == 0xFF) && (CAN1\_DATA\_buf[6] == 0xFF) && (CAN1\_DATA\_buf[7] == 0xFF)) {

FUELLEVELbits.value = CAN1\_DATA\_buf[1];

fmsData.fuelLevelValue = FUELLEVELbits.value;

//}

}

if (FUELLEVELbits.can\_number == 2 && (FUELLEVELbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

FUELLEVELbits.value = CAN2\_DATA\_buf[1];

fmsData.fuelLevelValue = FUELLEVELbits.value;

}

}

// Total Fuel Used - Общее израсходованное топливо

if (TOTALFUELUSEDCONFbits.can\_number != 0) {

if (TOTALFUELUSEDCONFbits.can\_number == 1 && (TOTALFUELUSEDCONFbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[0] == 0x00) && (CAN1\_DATA\_buf[1] == 0x00) && (CAN1\_DATA\_buf[2] == 0x00) && // ???

// (CAN1\_DATA\_buf[3] == 0x00)) {

TOTALFUELUSEDbits.value1 = CAN1\_DATA\_buf[4];

TOTALFUELUSEDbits.value2 = CAN1\_DATA\_buf[5];

TOTALFUELUSEDbits.value3 = CAN1\_DATA\_buf[6];

TOTALFUELUSEDbits.value4 = CAN1\_DATA\_buf[7];

fmsData.totalFuelUsedValue1 = TOTALFUELUSEDbits.value1;

fmsData.totalFuelUsedValue2 = TOTALFUELUSEDbits.value2;

fmsData.totalFuelUsedValue3 = TOTALFUELUSEDbits.value3;

fmsData.totalFuelUsedValue4 = TOTALFUELUSEDbits.value4;

//}

}

if (TOTALFUELUSEDCONFbits.can\_number == 2 && (TOTALFUELUSEDCONFbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

TOTALFUELUSEDbits.value1 = CAN2\_DATA\_buf[4];

TOTALFUELUSEDbits.value2 = CAN2\_DATA\_buf[5];

TOTALFUELUSEDbits.value3 = CAN2\_DATA\_buf[6];

TOTALFUELUSEDbits.value4 = CAN2\_DATA\_buf[7];

fmsData.totalFuelUsedValue1 = TOTALFUELUSEDbits.value1;

fmsData.totalFuelUsedValue2 = TOTALFUELUSEDbits.value2;

fmsData.totalFuelUsedValue3 = TOTALFUELUSEDbits.value3;

fmsData.totalFuelUsedValue4 = TOTALFUELUSEDbits.value4;

}

}

// High Resolution Total Vehicle Distance - Общий пробег

if (TOTALVEHICLEDISTANCECONFbits.can\_number != 0) {

if (TOTALVEHICLEDISTANCECONFbits.can\_number == 1 &&

(TOTALVEHICLEDISTANCECONFbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[4] == 0x80) && (CAN1\_DATA\_buf[5] == 0x31) && (CAN1\_DATA\_buf[6] == 0x03) && // ????

// (CAN1\_DATA\_buf[7] == 0x00)) {

TOTALVEHICLEDISTANCEbits.value1 = CAN1\_DATA\_buf[0];

TOTALVEHICLEDISTANCEbits.value2 = CAN1\_DATA\_buf[1];

TOTALVEHICLEDISTANCEbits.value3 = CAN1\_DATA\_buf[2];

TOTALVEHICLEDISTANCEbits.value4 = CAN1\_DATA\_buf[3];

fmsData.totalVehicleDistanceValue1 = TOTALVEHICLEDISTANCEbits.value1;

fmsData.totalVehicleDistanceValue2 = TOTALVEHICLEDISTANCEbits.value2;

fmsData.totalVehicleDistanceValue3 = TOTALVEHICLEDISTANCEbits.value3;

fmsData.totalVehicleDistanceValue4 = TOTALVEHICLEDISTANCEbits.value4;

//}

}

if (TOTALVEHICLEDISTANCECONFbits.can\_number == 2 &&

(TOTALVEHICLEDISTANCECONFbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

TOTALVEHICLEDISTANCEbits.value1 = CAN2\_DATA\_buf[0];

TOTALVEHICLEDISTANCEbits.value2 = CAN2\_DATA\_buf[1];

TOTALVEHICLEDISTANCEbits.value3 = CAN2\_DATA\_buf[2];

TOTALVEHICLEDISTANCEbits.value4 = CAN2\_DATA\_buf[3];

fmsData.totalVehicleDistanceValue1 = TOTALVEHICLEDISTANCEbits.value1;

fmsData.totalVehicleDistanceValue2 = TOTALVEHICLEDISTANCEbits.value2;

fmsData.totalVehicleDistanceValue3 = TOTALVEHICLEDISTANCEbits.value3;

fmsData.totalVehicleDistanceValue4 = TOTALVEHICLEDISTANCEbits.value4;

}

}

// Total Engine Hours - Общее количество моточасов

if (ENGINETOTALHOURSCONFbits.can\_number != 0) {

if (ENGINETOTALHOURSCONFbits.can\_number == 1 &&

(ENGINETOTALHOURSCONFbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[4] == 0x02) && (CAN1\_DATA\_buf[5] == 0x0C) && (CAN1\_DATA\_buf[6] == 0x00) && // ????

// (CAN1\_DATA\_buf[7] == 0x00)) {

ENGINETOTALHOURSbits.value1 = CAN1\_DATA\_buf[0];

ENGINETOTALHOURSbits.value2 = CAN1\_DATA\_buf[1];

ENGINETOTALHOURSbits.value3 = CAN1\_DATA\_buf[2];

ENGINETOTALHOURSbits.value4 = CAN1\_DATA\_buf[3];

fmsData.engineTotalHoursValue1 = ENGINETOTALHOURSbits.value1;

fmsData.engineTotalHoursValue2 = ENGINETOTALHOURSbits.value2;

fmsData.engineTotalHoursValue3 = ENGINETOTALHOURSbits.value3;

fmsData.engineTotalHoursValue4 = ENGINETOTALHOURSbits.value4;

//}

}

if (ENGINETOTALHOURSCONFbits.can\_number == 2 &&

(ENGINETOTALHOURSCONFbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

ENGINETOTALHOURSbits.value1 = CAN2\_DATA\_buf[0];

ENGINETOTALHOURSbits.value2 = CAN2\_DATA\_buf[1];

ENGINETOTALHOURSbits.value3 = CAN2\_DATA\_buf[2];

ENGINETOTALHOURSbits.value4 = CAN2\_DATA\_buf[3];

fmsData.engineTotalHoursValue1 = ENGINETOTALHOURSbits.value1;

fmsData.engineTotalHoursValue2 = ENGINETOTALHOURSbits.value2;

fmsData.engineTotalHoursValue3 = ENGINETOTALHOURSbits.value3;

fmsData.engineTotalHoursValue4 = ENGINETOTALHOURSbits.value4;

}

}

// Engine Coolant Temperature - Температура охлаждающей жидкости

if ( EGINECOOLANTTEMPERATUREbits.can\_number != 0) {

if (EGINECOOLANTTEMPERATUREbits.can\_number == 1 &&

(EGINECOOLANTTEMPERATUREbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[1] == 0xFF) && (CAN1\_DATA\_buf[2] == 0xFF) && (CAN1\_DATA\_buf[3] == 0xFF) && (CAN1\_DATA\_buf[4] == 0xFF) &&

// (CAN1\_DATA\_buf[5] == 0xFF) && (CAN1\_DATA\_buf[6] == 0xFF) && (CAN1\_DATA\_buf[7] == 0xFF)) { // ?????

EGINECOOLANTTEMPERATUREbits.value = CAN1\_DATA\_buf[0];

fmsData.engineCoolantTemperatureValue = EGINECOOLANTTEMPERATUREbits.value;

//}

}

if (EGINECOOLANTTEMPERATUREbits.can\_number == 2 &&

(EGINECOOLANTTEMPERATUREbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

EGINECOOLANTTEMPERATUREbits.value = CAN2\_DATA\_buf[0];

fmsData.engineCoolantTemperatureValue = EGINECOOLANTTEMPERATUREbits.value;

}

}

// Service Distance - Пробег до ТО

if (SERVICEDISTANCEbits.can\_number != 0) {

if (SERVICEDISTANCEbits.can\_number == 1 &&

(SERVICEDISTANCEbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[0] == 0xFF) && (CAN1\_DATA\_buf[3] == 0xFF) && (CAN1\_DATA\_buf[4] == 0xFF) &&

// (CAN1\_DATA\_buf[5] == 0xFF) && (CAN1\_DATA\_buf[6] == 0xFF) && (CAN1\_DATA\_buf[7] == 0xFF)) {

SERVICEDISTANCEbits.value1 = CAN1\_DATA\_buf[1];

SERVICEDISTANCEbits.value2 = CAN1\_DATA\_buf[2];

fmsData.serviceDistanceValue1 = SERVICEDISTANCEbits.value1;

fmsData.serviceDistanceValue2 = SERVICEDISTANCEbits.value2;

//}

}

if (SERVICEDISTANCEbits.can\_number == 2 &&

(SERVICEDISTANCEbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

SERVICEDISTANCEbits.value1 = CAN2\_DATA\_buf[1];

SERVICEDISTANCEbits.value2 = CAN2\_DATA\_buf[2];

fmsData.serviceDistanceValue1 = SERVICEDISTANCEbits.value1;

fmsData.serviceDistanceValue2 = SERVICEDISTANCEbits.value2;

}

}

// Engine Speed - Обороты двигателя

if (ENGINESPEEDbits.can\_number != 0) {

if (ENGINESPEEDbits.can\_number == 1 &&

(ENGINESPEEDbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[0] == 0xFF) && (CAN1\_DATA\_buf[1] == 0xFF) &&

// (CAN1\_DATA\_buf[5] == 0xFF) && (CAN1\_DATA\_buf[6] == 0xFF) && (CAN1\_DATA\_buf[7] == 0xFF)) {

ENGINESPEEDbits.value1 = CAN1\_DATA\_buf[3];

ENGINESPEEDbits.value2 = CAN1\_DATA\_buf[4];

fmsData.engineSpeedValue1 = ENGINESPEEDbits.value1;

fmsData.engineSpeedValue2 = ENGINESPEEDbits.value2;

//}

}

if (ENGINESPEEDbits.can\_number == 2 &&

(ENGINESPEEDbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

ENGINESPEEDbits.value1 = CAN2\_DATA\_buf[3];

ENGINESPEEDbits.value2 = CAN2\_DATA\_buf[4];

fmsData.engineSpeedValue1 = ENGINESPEEDbits.value1;

fmsData.engineSpeedValue2 = ENGINESPEEDbits.value2;

}

}

// Axle Weight - Нагрузка на оси

if (AXLEWEIGHTbits.can\_number != 0) {

if (AXLEWEIGHTbits.can\_number == 1 &&

(AXLEWEIGHTbits.id == (ID\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 1 );

//if ((CAN1\_DATA\_buf[3] == 0xFF) && (CAN1\_DATA\_buf[4] == 0xFF) &&

// (CAN1\_DATA\_buf[5] == 0xFF) && (CAN1\_DATA\_buf[6] == 0xFF) && (CAN1\_DATA\_buf[7] == 0xFF)) {

AXLEWEIGHTbits.value1 = CAN1\_DATA\_buf[1];

AXLEWEIGHTbits.value2 = CAN1\_DATA\_buf[2];

fmsData.axleWeightValue1 = AXLEWEIGHTbits.value1;

fmsData.axleWeightValue2 = AXLEWEIGHTbits.value2;

//}

}

if (AXLEWEIGHTbits.can\_number == 2 &&

(AXLEWEIGHTbits.id == (ID2\_tmp & maskIDExtendedFMS))) {

generate\_canbus\_sleep\_message ( 2 );

AXLEWEIGHTbits.value1 = CAN2\_DATA\_buf[1];

AXLEWEIGHTbits.value2 = CAN2\_DATA\_buf[2];

fmsData.axleWeightValue1 = AXLEWEIGHTbits.value1;

fmsData.axleWeightValue2 = AXLEWEIGHTbits.value2;

}

}

#endif

1. В функции init\_CAN\_filters (красным можно убрать, синим добавить):

#ifndef FOR\_TRUCK

if (CAN1SETTINGSbits.ide == 0) {

ecan1WriteRxAcptFilter( 15,(long)0x7EA, (unsigned int)CAN1SETTINGSbits.ide, 15, 0); //OBD

ecan1WriteRxAcptMask ( 0, 0x1FFFFFF0, 1, 0 );

} else {

ecan1WriteRxAcptFilter( 15,(long)0x18DAF128, (unsigned int)CAN1SETTINGSbits.ide, 15, 0); //OBD

ecan1WriteRxAcptMask ( 0, 0x1FFFFFE0, 1, 0 );

}

#endif

//#if defined(FOR\_TRUCK)

// CAN1SETTINGSbits.ide = 1; // extented - 29 bit

// ecan1WriteRxAcptFilter( 15,(long)maskIDExtendedLow, (unsigned int)CAN1SETTINGSbits.ide, 15, 0); //

// ecan1WriteRxAcptMask ( 1, maskIDExtendedHigh, 1, 1);

//#endif

// ecan1WriteRxAcptMask ( 3, 0x1FFFFFF0, 0, 0 );

/\* Enter Normal Mode \*/

C1CTRL1bits.REQOP=0;

while(C1CTRL1bits.OPMODE!=0);

/\* Request Configuration Mode \*/

C2CTRL1bits.REQOP=4;

while(C2CTRL1bits.OPMODE!=4);

if (CANIGNPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANIGNPRMSbits.filt\_number, (long)CANIGNPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANDDPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANDDPRMSbits.filt\_number, (long)CANDDPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANFPDPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANFPDPRMSbits.filt\_number, (long)CANFPDPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANRLDPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANRLDPRMSbits.filt\_number, (long)CANRLDPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANRRDPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANRRDPRMSbits.filt\_number, (long)CANRRDPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANBNPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANBNPRMSbits.filt\_number, (long)CANBNPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANBTPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANBTPRMSbits.filt\_number, (long)CANBTPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANODOPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANODOPRMSbits.filt\_number, (long)CANODOPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANFLPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANFLPRMSbits.filt\_number, (long)CANFLPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANBELTPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANBELTPRMSbits.filt\_number, (long)CANBELTPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (LOCKSTATPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)LOCKSTATPRMSbits.filt\_number, (long)LOCKSTATPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (LOCKSTATPRMS2bits.can\_number == 2) ecan2WriteRxAcptFilter( (int)LOCKSTATPRMS2bits.filt\_number,(long)LOCKSTATPRMS2bits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (CANIGNPRMS2bits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANIGNPRMS2bits.filt\_number, (long)CANIGNPRMS2bits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (WEBASTOSTATPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)WEBASTOSTATPRMSbits.filt\_number, (long)WEBASTOSTATPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (ALARMSTATPRMS2bits.can\_number == 2) ecan2WriteRxAcptFilter( (int)ALARMSTATPRMS2bits.filt\_number, (long)ALARMSTATPRMS2bits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (GEARBOXSTATPRMS2bits.can\_number == 2) ecan2WriteRxAcptFilter( (int)GEARBOXSTATPRMS2bits.filt\_number, (long)GEARBOXSTATPRMS2bits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (LOCKSTATPRMS3bits.can\_number == 2) ecan2WriteRxAcptFilter( (int)LOCKSTATPRMS3bits.filt\_number, (long)LOCKSTATPRMS3bits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //8 зажигание,

if (BOOTLOCKSTATPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)BOOTLOCKSTATPRMSbits.filt\_number, (long)BOOTLOCKSTATPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); //замок багажника тип 1

if (CANLOCKBYTESPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANLOCKBYTESPRMSbits.filt\_number, (long)CANLOCKBYTESPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANLIGHTSPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANLIGHTSPRMSbits.filt\_number, (long)CANLIGHTSPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANTLLIGHTSPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANTLLIGHTSPRMSbits.filt\_number, (long)CANTLLIGHTSPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANTRLIGHTSPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANTRLIGHTSPRMSbits.filt\_number, (long)CANTRLIGHTSPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANFLCNPNPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANFLCNPNPRMSbits.filt\_number, (long)CANFLCNPNPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANFL2PRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANFL2PRMSbits.filt\_number, (long)CANFL2PRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANBRPPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANBRPPRMSbits.filt\_number, (long)CANBRPPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (RPMPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)RPMPRMSbits.filt\_number, (long)RPMPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (TOYOTAKEYDATbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)TOYOTAKEYDATbits.filt\_number, (long)TOYOTAKEYDATbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

if (CANBUTTONPRMSbits.can\_number == 2) ecan2WriteRxAcptFilter( (int)CANBUTTONPRMSbits.filt\_number, (long)CANBUTTONPRMSbits.id, (unsigned int)CAN2SETTINGSbits.ide, 15, 0); // байты команд ключа для управления замками дверей

// ecan2WriteRxAcptMask ( 3, 0x1FFFFFF0, 1, 0 );

#if defined(FOR\_TRUCK)

CAN2SETTINGSbits.ide = 1; // extented - 29 bit

ecan2WriteRxAcptFilter( 15,(long)maskIDExtendedLow, (unsigned int)CAN1SETTINGSbits.ide, 15, 0); //

ecan2WriteRxAcptMask ( 1, maskIDExtendedHigh, 1, 1);

#endif

/\* Enter Normal Mode \*/

C2CTRL1bits.REQOP=0;

while(C2CTRL1bits.OPMODE!=0);

1. Функцию initTruckVariables можно просто заментить полность:

void initTruckVariables(void)

{

FUELLEVELbits.can\_number = 0x2;

FUELLEVELbits.value = 0x0;

FUELLEVELbits.id = 0x00FEFC00; //0x18FEFC21;

TOTALFUELUSEDCONFbits.can\_number = 0x2;

TOTALFUELUSEDCONFbits.id = 0x00FEE900; //0x18FEE927;

TOTALFUELUSEDbits.value1 = 0x0;

TOTALFUELUSEDbits.value2 = 0x0;

TOTALFUELUSEDbits.value3 = 0x0;

TOTALFUELUSEDbits.value4 = 0x0;

TOTALVEHICLEDISTANCECONFbits.can\_number = 0x2;

TOTALVEHICLEDISTANCECONFbits.id = 0x00FEC100; //0x18FEC1EE;

TOTALVEHICLEDISTANCEbits.value1 = 0x0;

TOTALVEHICLEDISTANCEbits.value2 = 0x0;

TOTALVEHICLEDISTANCEbits.value3 = 0x0;

TOTALVEHICLEDISTANCEbits.value4 = 0x0;

ENGINETOTALHOURSCONFbits.can\_number = 0x2;

ENGINETOTALHOURSCONFbits.id = 0x00FEE500; //0x18FEE527;

ENGINETOTALHOURSbits.value1 = 0x0;

ENGINETOTALHOURSbits.value2 = 0x0;

ENGINETOTALHOURSbits.value3 = 0x0;

ENGINETOTALHOURSbits.value4 = 0x0;

EGINECOOLANTTEMPERATUREbits.can\_number = 0x2;

EGINECOOLANTTEMPERATUREbits.id = 0x00FEEE00; //0x18FEEE00;

EGINECOOLANTTEMPERATUREbits.value = 0x0;

SERVICEDISTANCEbits.can\_number = 0x2;

SERVICEDISTANCEbits.id = 0x00FEC000; //0x18FEC027;

SERVICEDISTANCEbits.value1 = 0x0;

SERVICEDISTANCEbits.value2 = 0x0;

ENGINESPEEDbits.can\_number = 0x2;

ENGINESPEEDbits.id = 0x00F00400;

ENGINESPEEDbits.value1 = 0x0;

ENGINESPEEDbits.value2 = 0x0;

AXLEWEIGHTbits.can\_number = 0x2;

AXLEWEIGHTbits.id = 0x00FEEA00;

AXLEWEIGHTbits.value1 = 0x0;

AXLEWEIGHTbits.value2 = 0x0;

fmsData.axleWeightValue1 = 0x0;

fmsData.axleWeightValue2 = 0x0;

fmsData.engineSpeedValue1 = 0x0;

fmsData.engineSpeedValue2 = 0x0;

fmsData.serviceDistanceValue1 = 0x0;

fmsData.serviceDistanceValue2 = 0x0;

fmsData.engineTotalHoursValue1 = 0x0;

fmsData.engineTotalHoursValue2 = 0x0;

fmsData.engineTotalHoursValue3 = 0x0;

fmsData.engineTotalHoursValue4 = 0x0;

fmsData.totalVehicleDistanceValue1 = 0x0;

fmsData.totalVehicleDistanceValue2 = 0x0;

fmsData.totalVehicleDistanceValue3 = 0x0;

fmsData.totalVehicleDistanceValue4 = 0x0;

fmsData.totalFuelUsedValue1 = 0x0;

fmsData.totalFuelUsedValue2 = 0x0;

fmsData.totalFuelUsedValue3 = 0x0;

fmsData.totalFuelUsedValue4 = 0x0;

fmsData.engineCoolantTemperatureValue = 0x0;

fmsData.fuelLevelValue = 0x0;

}