1. Код сортировки (файл SORTING):

void \_INIT ProgramInit(void)

{

BR\_HEIGHT\_1 = 0;

BR\_HEIGHT\_2 = 0;

}

void \_CYCLIC ProgramCyclic(void)

{

if (crane\_1\_overstock != 1)

{

if((BR\_HEIGHT\_1 == 1)&&(BR\_HEIGHT\_2 == 1))

{

if(cnv\_m == 1)

{

BR\_ARM\_TURN\_S = 0;

BR\_ARM\_TURN\_M = 1;

BR\_ARM\_GO\_S = 0;

BR\_ARM\_GO\_M = 1;

}

else

{

BR\_ARM\_TURN\_S = 0;

BR\_ARM\_TURN\_M = 0;

BR\_ARM\_GO\_S = 0;

BR\_ARM\_GO\_M = 0;

}

}

if((BR\_HEIGHT\_1 == 1)&&(BR\_HEIGHT\_2 == 0))

{

if(cnv\_s == 1)

{

BR\_ARM\_TURN\_S = 1;

BR\_ARM\_TURN\_M = 0;

BR\_ARM\_GO\_S = 1;

BR\_ARM\_GO\_M = 0;

}

else

{

BR\_ARM\_TURN\_S = 0;

BR\_ARM\_TURN\_M = 0;

BR\_ARM\_GO\_S = 0;

BR\_ARM\_GO\_M = 0;

}

}

}

else

{

BR\_ARM\_TURN\_S = 0;

BR\_ARM\_TURN\_M = 0;

BR\_ARM\_GO\_S = 0;

BR\_ARM\_GO\_M = 0;

}

}

1. Код Palletizer 1 (файл PLT\_1):

void \_INIT ProgramInit(void)  
{  
 i=0;  
 go=0;  
 BR\_PLT\_TURN\_1=1;  
 BR\_PLT\_BELT\_1=1;  
 cnv\_s=1;  
 BR\_PLT\_CNV\_1=1;  
 BR\_PLT\_GO\_1=1;  
 BR\_PLT\_UP\_1=0;  
 BR\_PLT\_DOWN\_1=0;  
 k=0;  
 BR\_PLT\_EL\_FULL\_1 = 1;  
  
}  
  
void \_CYCLIC ProgramCyclic(void)  
**{**   
 if(BR\_RESET == 1)  
 {  
 i=0;  
 go=0;  
 BR\_PLT\_TURN\_1=1;  
 BR\_PLT\_BELT\_1=1;  
 cnv\_s=1;  
 BR\_PLT\_CNV\_1=1;  
 BR\_PLT\_GO\_1=1;  
 BR\_PLT\_UP\_1=0;  
 BR\_PLT\_DOWN\_1=0;  
 k=0;  
 BR\_PLT\_EL\_FULL\_1 = 1;  
 BR\_PLT\_PUSH\_1 = 0;  
 BR\_PLT\_CLAMP\_1 = 0;  
 }  
   
 if((BR\_PLT\_FRONT\_1 == 1) && (BR\_PLT\_BACK\_1 == 1) && (k==0)) //условие для поднятия палеты  
 {  
 BR\_PLT\_CNV\_1=0;  
 BR\_PLT\_GO\_1=0;  
 BR\_PLT\_EL\_FULL\_1 = 1;  
 BR\_PLT\_UP\_1 = 1;  
 k++;  
 }  
 switch (k) // поднимаем палету  
 {  
 case (1):  
 {  
 if (BR\_PLT\_EL\_MOVING\_1 == 0)  
 {  
 k++;  
 }  
 }  
 break;  
 case (2):  
 {  
 if (BR\_PLT\_EL\_MOVING\_1 == 1)  
 {  
 k++;  
 }  
 }  
 break;  
 case (3):  
 {  
 if (BR\_PLT\_EL\_MOVING\_1 == 0)  
 {  
 BR\_PLT\_UP\_1 = 0;  
 k++;  
 }  
 }  
 break;   
 } //поднимаем палету  
  
 if((BR\_PLT\_SENS\_1 == 1) && (i==0) && (go<2)) //запускаем палету к крану либо продолжаем заполнять слои на палетах  
 {  
 BR\_PLT\_BELT\_1=1;  
 cnv\_s=1;  
 i++;  
 }  
   
 switch (i) // считаем кол-во коробок  
 {  
 case (1):  
 {  
 if(BR\_PLT\_SENS\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (3):  
 {  
 if(BR\_PLT\_SENS\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (5):  
 {  
 if(BR\_PLT\_SENS\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (11):  
 {  
 if(BR\_PLT\_SENS\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (13):  
 {  
 if(BR\_PLT\_SENS\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (15):  
 {  
 if(BR\_PLT\_SENS\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (2):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (4):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (6):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (12):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (14):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (16):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 } //считаем кол-во коробок  
   
 switch (i) // останавливаем/запускаем конвейеры  
 {  
 case (7):  
 {  
 cnv\_s=0;  
 BR\_PLT\_BELT\_1 = 0;  
 i++;  
 }  
 break;  
 case (10):  
 {  
 if(BR\_PLT\_SENS\_1 == 1)  
 {  
 cnv\_s=1;  
 BR\_PLT\_BELT\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 case (17):  
 {  
 cnv\_s=0;  
 BR\_PLT\_BELT\_1 = 0;  
 i++;  
 }  
 break;  
 } //останавливаем/запускаем конвейеры  
   
 switch (i) //толкаем коробки  
 {  
 case (8):  
 {  
 if (BR\_PLT\_PUSH\_DID\_1 == 1)  
 {  
 BR\_PLT\_PUSH\_1 = 1;  
 }  
 if (BR\_PLT\_PUSH\_DID\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (9):  
 {  
 if (BR\_PLT\_PUSH\_DID\_1 == 1)  
 {  
 BR\_PLT\_PUSH\_1 = 0;  
 i++;  
 }  
 }  
 break;  
 case (18):  
 {  
 if (BR\_PLT\_PUSH\_DID\_1 == 1)  
 {  
 BR\_PLT\_PUSH\_1 = 1;  
 }  
 if (BR\_PLT\_PUSH\_DID\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (19):  
 {  
 if (BR\_PLT\_PUSH\_DID\_1 == 1)  
 {  
 BR\_PLT\_PUSH\_1 = 0;  
 i++;  
 }  
 }  
 break;  
 } //толкаем коробки  
   
 switch (i) //фиксируем и тромбуем  
 {  
 case (20):  
 {  
 if (BR\_PLT\_CLAMP\_DID\_1 == 0)  
 {  
 BR\_PLT\_CLAMP\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 case (21):  
 {  
 if (BR\_PLT\_CLAMP\_DID\_1 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 } //фиксируем и тромбуем  
   
 switch (i) //опускаем на палеты  
 {  
 case(22):  
 {  
 if(BR\_PLT\_PLATE\_DID\_1 == 1)  
 {  
 BR\_PLT\_PLATE\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 case(23):  
 {  
 if(BR\_PLT\_PLATE\_DID\_1 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case(24):  
 {  
 if(BR\_PLT\_PLATE\_DID\_1 == 1)  
 {  
 BR\_PLT\_EL\_FULL\_1 = 0;  
 BR\_PLT\_DOWN\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 } //опускаем на палеты  
   
 switch (i)// перезапускаем цикл  
 {  
 case(25):  
 {  
 if(BR\_PLT\_EL\_MOVING\_1 == 0)  
 {  
 i++;  
 }   
 }  
 break;  
 case(26):  
 {  
 if(BR\_PLT\_EL\_MOVING\_1 == 1)  
 {  
 i++;  
 }   
 }  
 break;  
 case(27):  
 {  
 if(BR\_PLT\_EL\_MOVING\_1 == 0)  
 {  
 BR\_PLT\_CLAMP\_1 = 0;  
 BR\_PLT\_PLATE\_1 = 0;  
 BR\_PLT\_DOWN\_1 = 0;  
 i=0;  
 go++;  
 }   
 }  
 break;  
 } //перезапускаем цикл  
  
 switch (go) //запускаем палету после укомплектования  
 {  
 case(2):  
 {  
 BR\_PLT\_EL\_FULL\_1 = 1;  
 go++;  
 }  
 break;  
 case(3):  
 {  
 BR\_PLT\_DOWN\_1 = 1;  
 go++;;  
 }  
 break;  
 case(4):  
 {  
 if(BR\_PLT\_EL\_MOVING\_1 == 1)  
 {  
 go++;  
 }  
 }  
 break;  
 case(5):  
 {  
 if(BR\_PLT\_EL\_MOVING\_1 == 0)  
 {  
 BR\_PLT\_GO\_1 = 1;  
 BR\_PLT\_DOWN\_1 = 0;  
 go++;  
 }  
 }  
 break;  
 } //запускаем палету после укомплектования   
   
 if((BR\_PLT\_FRONT\_1 == 1) && (BR\_PLT\_BACK\_1 == 0)) //условие для запуска подводящего палеты конвейера  
 {  
 BR\_PLT\_CNV\_1=1;  
 k=0;  
 }  
**}**

1. Код Palletizer 2 (файл PLT\_2):

void \_INIT ProgramInit(void)  
{  
 i=0;  
 go1=0;  
 BR\_PLT\_TURN\_2=1;  
 BR\_PLT\_BELT\_2=1;  
 cnv\_m=1;  
 BR\_PLT\_CNV\_2=1;  
 BR\_PLT\_GO\_2=1;  
 BR\_PLT\_UP\_2=0;  
 BR\_PLT\_DOWN\_2=0;  
 k=0;  
 BR\_PLT\_EL\_FULL\_2 = 1;  
  
}  
  
void \_CYCLIC ProgramCyclic(void)  
**{**   
 if(BR\_RESET == 1)  
 {  
 i=0;  
 go1=0;  
 BR\_PLT\_TURN\_2=1;  
 BR\_PLT\_BELT\_2=1;  
 cnv\_m=1;  
 BR\_PLT\_CNV\_2=1;  
 BR\_PLT\_GO\_2=1;  
 BR\_PLT\_UP\_2=0;  
 BR\_PLT\_DOWN\_2=0;  
 k=0;  
 BR\_PLT\_EL\_FULL\_2 = 1;  
 BR\_PLT\_PUSH\_2 = 0;  
 BR\_PLT\_CLAMP\_2 = 0;  
 }  
 if((BR\_PLT\_FRONT\_2 == 1) && (BR\_PLT\_BACK\_2 == 1) && (k==0)) //условие для поднятия палеты  
 {  
 BR\_PLT\_CNV\_2=0;  
 BR\_PLT\_GO\_2=0;  
 BR\_PLT\_EL\_FULL\_2 = 1;  
 BR\_PLT\_UP\_2 = 1;  
 k++;  
 }  
   
 switch (k) // поднимаем палету  
 {  
 case (1):  
 {  
 if (BR\_PLT\_EL\_MOVING\_2 == 0)  
 {  
 k++;  
 }  
 }  
 break;  
 case (2):  
 {  
 if (BR\_PLT\_EL\_MOVING\_2 == 1)  
 {  
 k++;  
 }  
 }  
 break;  
 case (3):  
 {  
 if (BR\_PLT\_EL\_MOVING\_2 == 0)  
 {  
 BR\_PLT\_UP\_2 = 0;  
 k++;  
 }  
 }  
 break;   
 } //поднимаем палету  
   
 if((BR\_PLT\_SENS\_2 == 1) && (i==0) && (go1<1)) //запускаем палету к крану либо продолжаем заполнять слои на палетах  
 {  
 BR\_PLT\_BELT\_2=1;  
 cnv\_m=1;  
 i++;  
 }  
  
 switch (i) // считаем кол-во коробок  
 {  
 case (1):  
 {  
 if(BR\_PLT\_SENS\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (2):  
 {  
 if(BR\_PLT\_SENS\_2 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (7):  
 {  
 if(BR\_PLT\_SENS\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (8):  
 {  
 if(BR\_PLT\_SENS\_2 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (13):  
 {  
 if(BR\_PLT\_SENS\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (14):  
 {  
 if(BR\_PLT\_SENS\_2 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 } //считаем кол-во коробок  
   
 switch (i) // останавливаем/запускаем конвейеры  
 {  
 case (3):  
 {  
 cnv\_m=0;  
 BR\_PLT\_BELT\_2 = 0;  
 i++;  
 }  
 break;  
 case (6):  
 {  
 if(BR\_PLT\_SENS\_2 == 1)  
 {  
 cnv\_m=1;  
 BR\_PLT\_BELT\_2 = 1;  
 i++;  
 }  
 }  
 break;  
 case (9):  
 {  
 cnv\_m=0;  
 BR\_PLT\_BELT\_2 = 0;  
 i++;  
 }  
 break;  
 case (12):  
 {  
 if(BR\_PLT\_SENS\_2 == 1)  
 {  
 cnv\_m=1;  
 BR\_PLT\_BELT\_2 = 1;  
 i++;  
 }  
 }  
 break;  
 case (15):  
 {  
 cnv\_m=0;  
 BR\_PLT\_BELT\_2 = 0;  
 i++;  
 }  
 break;  
 case (18):  
 {  
 if(BR\_PLT\_SENS\_2 == 1)  
 {  
 cnv\_m=1;  
 BR\_PLT\_BELT\_2 = 1;  
 i++;  
 }  
 }  
 break;  
 } //останавливаем/запускаем конвейеры  
   
 switch (i) //толкаем коробки  
 {  
 case (4):  
 {  
 if (BR\_PLT\_PUSH\_DID\_2 == 1)  
 {  
 BR\_PLT\_PUSH\_2 = 1;  
 }  
 if (BR\_PLT\_PUSH\_DID\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (5):  
 {  
 if (BR\_PLT\_PUSH\_DID\_2 == 1)  
 {  
 BR\_PLT\_PUSH\_2 = 0;  
 i++;  
 }  
 }  
 break;  
 case (10):  
 {  
 if (BR\_PLT\_PUSH\_DID\_2 == 1)  
 {  
 BR\_PLT\_PUSH\_2 = 1;  
 }  
 if (BR\_PLT\_PUSH\_DID\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (11):  
 {  
 if (BR\_PLT\_PUSH\_DID\_2 == 1)  
 {  
 BR\_PLT\_PUSH\_2 = 0;  
 i++;  
 }  
 }  
 break;  
 case (16):  
 {  
 if (BR\_PLT\_PUSH\_DID\_2 == 1)  
 {  
 BR\_PLT\_PUSH\_2 = 1;  
 }  
 if (BR\_PLT\_PUSH\_DID\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case (17):  
 {  
 if (BR\_PLT\_PUSH\_DID\_2 == 1)  
 {  
 BR\_PLT\_PUSH\_2 = 0;  
 i++;  
 }  
 }  
 break;  
 } //толкаем коробки  
   
 switch (i) //фиксируем и тромбуем  
 {  
 case (19):  
 {  
 if (BR\_PLT\_CLAMP\_DID\_2 == 0)  
 {  
 cnv\_m=0;  
 BR\_PLT\_BELT\_2 = 0;  
 BR\_PLT\_CLAMP\_2 = 1;  
 i++;  
 }  
 }  
 break;  
 case (20):  
 {  
 if (BR\_PLT\_CLAMP\_DID\_2 == 1)  
 {  
 i++;  
 }  
 }  
 break;  
 } //фиксируем и тромбуем  
   
 switch (i) //опускаем на палеты  
 {  
 case(21):  
 {  
 if(BR\_PLT\_PLATE\_DID\_2 == 1)  
 {  
 BR\_PLT\_PLATE\_2 = 1;  
 i++;  
 }  
 }  
 break;  
 case(22):  
 {  
 if(BR\_PLT\_PLATE\_DID\_2 == 0)  
 {  
 i++;  
 }  
 }  
 break;  
 case(23):  
 {  
 if(BR\_PLT\_PLATE\_DID\_2 == 1)  
 {  
 BR\_PLT\_EL\_FULL\_2 = 0;  
 BR\_PLT\_DOWN\_2 = 1;  
 i++;  
 }  
 }  
 break;  
 case (24):  
 {  
 BR\_PLT\_DOWN\_2 = 0;  
 i++;  
 }  
 break;  
 case (25):  
 {  
 BR\_PLT\_DOWN\_2 = 1;  
 i++;  
 }  
 break;  
 } //опускаем на палеты  
  
 switch (i)// перезапускаем цикл  
 {  
 case(26):  
 {  
 if(BR\_PLT\_EL\_MOVING\_2 == 1)  
 {  
 i++;  
 }   
 }  
 break;  
 case(27):  
 {  
 if(BR\_PLT\_EL\_MOVING\_2 == 0)  
 {  
 BR\_PLT\_CLAMP\_2 = 0;  
 BR\_PLT\_PLATE\_2 = 0;  
 BR\_PLT\_DOWN\_2 = 0;  
 i=0;  
 go1++;  
 }   
 }  
 break;  
 } //перезапускаем цикл  
  
 switch (go1) //запускаем палету после укомплектования  
 {  
 case(1):  
 {  
 BR\_PLT\_EL\_FULL\_2 = 1;  
 go1++;  
 }  
 break;  
 case(2):  
 {  
 BR\_PLT\_DOWN\_2 = 1;  
 go1++;;  
 }  
 break;  
 case(3):  
 {  
 if(BR\_PLT\_EL\_MOVING\_2 == 1)  
 {  
 go1++;  
 }  
 }  
 break;  
 case(4):  
 {  
 if(BR\_PLT\_EL\_MOVING\_2 == 0)  
 {  
 BR\_PLT\_GO\_2 = 1;  
 BR\_PLT\_DOWN\_2 = 0;  
 go1++;  
 }  
 }  
 break;  
 } //запускаем палету после укомплектования   
  
 if((BR\_PLT\_FRONT\_2 == 1) && (BR\_PLT\_BACK\_2 == 0)) //условие для запуска подводящего палеты конвейера  
 {  
 BR\_PLT\_CNV\_2=1;  
 k=0;  
 }  
**}**

1. Код подводящих конвейеров (файл CONVEYORS):

void \_CYCLIC ProgramCyclic(void)  
**{** if(cnv\_s == 0)  
 {  
 BR\_CNV\_S\_1 = 0;  
 BR\_CNV\_S\_2 = 0;  
 BR\_CNV\_S\_3 = 0;  
 BR\_CNV\_S\_4 = 0;  
 BR\_CNV\_S\_5 = 0;  
 }  
 else  
 {  
 BR\_CNV\_S\_1 = 1;  
 BR\_CNV\_S\_2 = 1;  
 BR\_CNV\_S\_3 = 1;  
 BR\_CNV\_S\_4 = 1;  
 BR\_CNV\_S\_5 = 1;  
 }  
 if(cnv\_m == 0)  
 {  
 BR\_CNV\_M\_1 = 0;  
 BR\_CNV\_M\_2 = 0;  
 BR\_CNV\_M\_3 = 0;  
 BR\_CNV\_M\_4 = 0;  
 BR\_CNV\_M\_5 = 0;  
 }  
 else  
 {  
 BR\_CNV\_M\_1 = 1;  
 BR\_CNV\_M\_2 = 1;  
 BR\_CNV\_M\_3 = 1;  
 BR\_CNV\_M\_4 = 1;  
 BR\_CNV\_M\_5 = 1;  
 }  
**}**

1. Код крана погрузки 1 (файл CRANE\_1):

void \_INIT ProgramInit(void)  
{  
 i=0;  
 n=0;  
}  
  
void \_CYCLIC ProgramCyclic(void)  
**{** if(BR\_RESET == 1)  
 {  
 i=0;  
 n=0;  
 BR\_CRANE\_CONV\_1 = 0;  
 BR\_CRANE\_RIGHT\_1 = 0;  
 BR\_CRANE\_LIFT\_1 = 0;  
 BR\_CRANE\_TG\_POS\_1 = 55;  
 BR\_CRANE\_LIFT\_1 = 0;  
 go = 0;  
 }  
 if((go==6)&(i==0))  
 {  
 BR\_CRANE\_CONV\_1 = 1;  
 go=0;  
 i++;  
 }  
   
 switch (i)  
 {  
 case (1):  
 {  
 if(BR\_CRANE\_SENS\_1==1)  
 {  
 BR\_CRANE\_CONV\_1 = 0;  
 BR\_CRANE\_LEFT\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 case (2):  
 {  
 if(BR\_CRANE\_LIM\_LEFT\_1==1)  
 {  
 BR\_CRANE\_LIFT\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 case (3):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_1==1)  
 {  
 BR\_CRANE\_LEFT\_1 = 0;  
 i++;  
 }  
 }  
 break;  
 case (4):  
 {  
 if(BR\_CRANE\_LIM\_MIDDLE\_1==1)  
 {  
 while((crane\_1\_space[n]!= 0) && (n<55))  
 {  
 n++;  
 }  
 if (n!=55)  
 {  
 crane\_1\_space[n] = 1;  
 BR\_CRANE\_TG\_POS\_1 = (n+1);  
 n=0;  
 i++;  
 }  
 else  
 {  
 crane\_1\_overstock = 1;  
 n=0;  
 i++;  
 }  
 }  
 }  
 break;  
 case (5):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_1==1)&&(BR\_CRANE\_MOVING\_X\_1==1) && (crane\_1\_overstock != 1))  
 {  
 i++;  
 }  
 }  
 break;  
 case (6):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_1==0)&&(BR\_CRANE\_MOVING\_X\_1==0))  
 {  
 BR\_CRANE\_RIGHT\_1 = 1;  
 i++;  
 }  
 }  
 break;  
 case (7):  
 {  
 if(BR\_CRANE\_LIM\_RIGHT\_1==1)  
 {  
 BR\_CRANE\_LIFT\_1 = 0;  
 i++;  
 }  
 }  
 break;  
 case (8):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_1==1)  
 {  
 i++;  
 }  
 }  
 break;  
 case (9):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_1==1)  
 {  
 BR\_CRANE\_RIGHT\_1 = 0;  
 i++;  
 }  
 }  
 break;  
 case (10):  
 {  
 if(BR\_CRANE\_LIM\_MIDDLE\_1==1)  
 {  
 BR\_CRANE\_TG\_POS\_1 = 55;  
 i=0;  
 }  
 }  
 break;  
 }  
**}**

1. Код крана погрузки 2 (файл CRANE\_2):

void \_INIT ProgramInit(void)  
{  
 i=0;  
 n=0;  
}  
  
void \_CYCLIC ProgramCyclic(void)  
**{** if(BR\_RESET == 1)  
 {  
 i=0;  
 n=0;  
 BR\_CRANE\_CONV\_2 = 0;  
 BR\_CRANE\_RIGHT\_2 = 0;  
 BR\_CRANE\_LIFT\_2 = 0;  
 BR\_CRANE\_TG\_POS\_2 = 55;  
 BR\_CRANE\_LIFT\_2 = 0;  
 go1 = 0;  
 }  
 if((go1==5)&(i==0))  
 {  
 BR\_CRANE\_CONV\_2 = 1;  
 go1=0;  
 i++;  
 }  
 if((i==1)&&(BR\_CRANE\_SENS\_2==1))  
 {  
 BR\_CRANE\_CONV\_2 = 0;  
 BR\_CRANE\_RIGHT\_2 = 1;  
 i++;  
 }  
 if((i==2)&&(BR\_CRANE\_LIM\_RIGHT\_2==1))  
 {  
 BR\_CRANE\_LIFT\_2 = 1;  
 i++;  
 }  
 if((i==3)&&(BR\_CRANE\_MOVING\_Z\_2==1))  
 {  
 BR\_CRANE\_RIGHT\_2 = 0;  
 i++;  
 }  
 if((i==4)&&(BR\_CRANE\_LIM\_MIDDLE\_2==1))  
 {  
 while((crane\_2\_space[n]!= 0) && (n<54))  
 {  
 n++;  
 }  
 if (n!=54)  
 {  
 crane\_2\_space[n] = 1;  
 BR\_CRANE\_TG\_POS\_2 = (n+1);  
 n=0;  
 i++;  
 }  
 else  
 {  
 crane\_2\_overstock = 1;  
 n=0;  
 i++;  
 }  
 }  
 if((i==5)&&(BR\_CRANE\_MOVING\_Z\_2==1)&&(BR\_CRANE\_MOVING\_X\_2==1) && (crane\_2\_overstock != 1))  
 {  
 i++;  
 }  
 if((i==6)&&(BR\_CRANE\_MOVING\_Z\_2==0)&&(BR\_CRANE\_MOVING\_X\_2==0))  
 {  
 BR\_CRANE\_LEFT\_2 = 1;  
 i++;  
 }  
 if((i==7)&&(BR\_CRANE\_LIM\_LEFT\_2==1))  
 {  
 BR\_CRANE\_LIFT\_2 = 0;  
 i++;  
 }  
 if((i==8)&&(BR\_CRANE\_MOVING\_Z\_2==1))  
 {  
 i++;  
 }  
 if((i==9)&&(BR\_CRANE\_MOVING\_Z\_2==0))  
 {  
 BR\_CRANE\_LEFT\_2 = 0;  
 i++;  
 }  
 if((i==10)&&(BR\_CRANE\_LIM\_MIDDLE\_2==1))  
 {  
 BR\_CRANE\_TG\_POS\_2 = 55;  
 i=0;  
 }  
**}**

1. Код крана разгрузки (файл CRANE\_EXIT):

void \_INIT ProgramInit(void)  
{  
 i=0;  
 n=0;  
}  
  
void \_CYCLIC ProgramCyclic(void)  
**{** if(BR\_RESET == 1)  
 {  
 i=0;  
 n=0;  
 BR\_CRANE\_CONV\_E = 0;  
 BR\_CRANE\_RIGHT\_E = 0;  
 BR\_CRANE\_LIFT\_E = 0;  
 BR\_CRANE\_TG\_POS\_E = 55;  
 BR\_CRANE\_LIFT\_E = 0;  
 go1 = 0;  
 }  
   
  
 if(((BR\_CRANE\_EXIT\_2\_E == 1) || (BR\_CRANE\_EXIT\_2\_E\_LIGHT == 1)) && (BR\_CRANE\_EXIT\_1\_E\_LIGHT == 0))  
 {   
 n=0;  
 switch(i)  
 {  
 case(0):  
 {  
 while((crane\_2\_space[n] != 1) && (n<54))  
 {  
 n++;  
 }  
 if ((n!=54) && (crane\_2\_space[n] == 1))  
 {  
 crane\_2\_space[n] = 0;  
 BR\_CRANE\_TG\_POS\_E = (n+1);  
 n=0;  
 i++;  
 BR\_CRANE\_EXIT\_2\_E\_LIGHT = 1;  
 }  
 }  
 break;  
   
 case(1):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==1)&&(BR\_CRANE\_MOVING\_X\_E==1))  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(2):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==0)&&(BR\_CRANE\_MOVING\_X\_E==0))  
 {  
 BR\_CRANE\_RIGHT\_E = 1;  
 i++;  
 }  
 }  
 break;   
   
 case(3):  
 {  
 if(BR\_CRANE\_LIM\_RIGHT\_E==1)  
 {  
 BR\_CRANE\_LIFT\_E = 1;  
 i++;  
 }  
 }  
 break;  
   
 case(4):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==1)  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(5):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==0)  
 {  
 BR\_CRANE\_RIGHT\_E = 0;  
 i++;  
 }  
 }  
 break;  
   
 case(6):  
 {  
 if(BR\_CRANE\_LIM\_MIDDLE\_E==1)  
 {  
 BR\_CRANE\_TG\_POS\_E = 55;  
 i++;  
 }  
 }  
 break;  
   
 case(7):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==1)&&(BR\_CRANE\_MOVING\_X\_E==1))  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(8):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==0)&&(BR\_CRANE\_MOVING\_X\_E==0))  
 {  
 BR\_CRANE\_RIGHT\_E = 1;  
 i++;  
 }  
 }  
 break;  
   
 case(9):  
 {  
 if(BR\_CRANE\_LIM\_RIGHT\_E==1)  
 {  
 BR\_CRANE\_LIFT\_E = 0;  
 i++;  
 }  
 }  
 break;  
   
 case(10):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==1)  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(11):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==0)  
 {  
 BR\_CRANE\_RIGHT\_E = 0;  
 i=0;  
 BR\_CRANE\_EXIT\_2\_E\_LIGHT = 0;  
 }  
 }  
 break;  
 }   
 }  
   
 if(((BR\_CRANE\_EXIT\_1\_E == 1) || (BR\_CRANE\_EXIT\_1\_E\_LIGHT == 1)) && (BR\_CRANE\_EXIT\_2\_E\_LIGHT == 0))  
 {   
 n=0;  
 switch(i)  
 {  
 case(0):  
 {  
 while((crane\_1\_space[n]!= 1) && (n<54))  
 {  
 n++;  
 }  
 if ((n!=54) && (crane\_1\_space[n] == 1))  
 {  
 crane\_1\_space[n] = 0;  
 BR\_CRANE\_TG\_POS\_E = (n+1);  
 n=0;  
 i++;  
 BR\_CRANE\_EXIT\_1\_E\_LIGHT = 1;  
 }  
 }  
 break;  
   
 case(1):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==1)&&(BR\_CRANE\_MOVING\_X\_E==1))  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(2):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==0)&&(BR\_CRANE\_MOVING\_X\_E==0))  
 {  
 BR\_CRANE\_LEFT\_E = 1;  
 i++;  
 }  
 }  
 break;  
   
 case(3):  
 {  
 if(BR\_CRANE\_LIM\_LEFT\_E==1)  
 {  
 BR\_CRANE\_LIFT\_E = 1;  
 i++;  
 }  
 }  
 break;  
   
 case(4):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==1)  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(5):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==0)  
 {  
 BR\_CRANE\_LEFT\_E = 0;  
 i++;  
 }  
 }  
 break;  
   
 case(6):  
 {  
 if(BR\_CRANE\_LIM\_MIDDLE\_E==1)  
 {  
 BR\_CRANE\_TG\_POS\_E = 55;  
 i++;  
 }  
 }  
 break;  
   
 case(7):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==1)&&(BR\_CRANE\_MOVING\_X\_E==1))  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(8):  
 {  
 if((BR\_CRANE\_MOVING\_Z\_E==0)&&(BR\_CRANE\_MOVING\_X\_E==0))  
 {  
 BR\_CRANE\_RIGHT\_E = 1;  
 i++;  
 }  
 }  
 break;  
   
 case(9):  
 {  
 if(BR\_CRANE\_LIM\_RIGHT\_E==1)  
 {  
 BR\_CRANE\_LIFT\_E = 0;  
 i++;  
 }  
 }  
 break;  
   
 case(10):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==1)  
 {  
 i++;  
 }  
 }  
 break;  
   
 case(11):  
 {  
 if(BR\_CRANE\_MOVING\_Z\_E==0)  
 {  
 BR\_CRANE\_RIGHT\_E = 0;  
 i=0;  
 BR\_CRANE\_EXIT\_1\_E\_LIGHT = 0;  
 }  
 }  
 break;  
 }   
 }  
**}**