# BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 10 (P2)

## Assignment 1

- **Vẽ tam giác:**

- Code

```
.eqv HEADING 0xffff8010 # Integer: Goc quay tu 0 den 359
# 0 : Tren
# 90: Phai
# 180: Duoi
# 270: Trai
.eqv MOVING 0xffff8050 # Boolean: Co di chuyen hay khong
.eqv LEAVETRACK 0xffff8020 # Boolean (0 hoac !0):
# Co track hay khong
.eqv WHEREX 0xffff8030 # Integer: Doc gia tri X hien tai cua con bot
.eqv WHEREY 0xffff8040 # Integer: Doc gia tri Y hien tai cua con bot
.text
main:
addi $a0, $0, 90 # Quay sang trai de bat dau chay
jal ROTATE
jal GO
sleep0:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
jal STOP

jal TRACK # Danh dau dia diem hien tai de bat dau ve canh 1
```

```
addi $a0, $0, 150 # Quay goc 150
jal ROTATE
jal GO
sleep1:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau TRACK de bat dau ve canh 2
addi $a0, $0, 30 # Quay goc 30
jal ROTATE
jal GO
sleep2:
addi $v0, $zero, 32 # De no chay trong 2990ms
li $a0, 2990        # cho bot chay gan den goc do thi dung
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau TRACK
addi $a0, $0, 270 # Quay goc
jal ROTATE
jal GO
sleep3:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
```

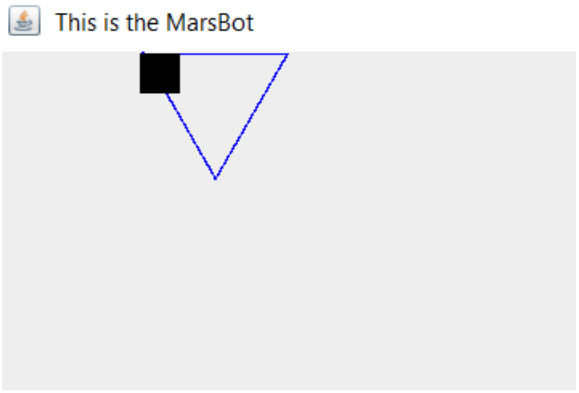
```
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

#jal TRACK # Danh dau TRACK
jal STOP
endmain:
li $v0, 10
syscall

GO:
li $at, MOVING # Thay doi cong MOVING
addi $k0, $0, 1 # logic 1
sb $k0, ($at) # Bat dau chay
jr $ra
STOP:
li $at, MOVING # Thay doi cong MOVING
sb $0, ($at) # Dung chay
jr $ra
TRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
addi $k0, $0, 1 # logic 1
sb $k0, 0($at) # bat dau tracking
jr $ra
UNTRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
sb $0, 0($at) # dung ve
jr $ra
ROTATE:
```

li $at, HEADING # Thay doi cong HEAD

sw $a0, ($at) # Xoay robot

jr $ra

- Kết quả:




- **Vẽ hình vuông:**

- Code:

.eqv HEADING 0xffff8010 # Integer: Goc quay tu 0 den 359

# 0 : Tren

# 90: Phai

# 180: Duoi

# 270: Trai

.eqv MOVING 0xffff8050 # Boolean: Co di chuyen hay khong

.eqv LEAVETRACK 0xffff8020 # Boolean (0 hoac !0):

# Co track hay khong

.eqv WHEREX 0xffff8030 # Integer: Doc gia tri X hien tai cua con bot

.eqv WHEREY 0xffff8040 # Integer: Doc gia tri Y hien tai cua con bot

.text

main:

addi $a0, $0, 90 # Quay sang trai de bat dau chay

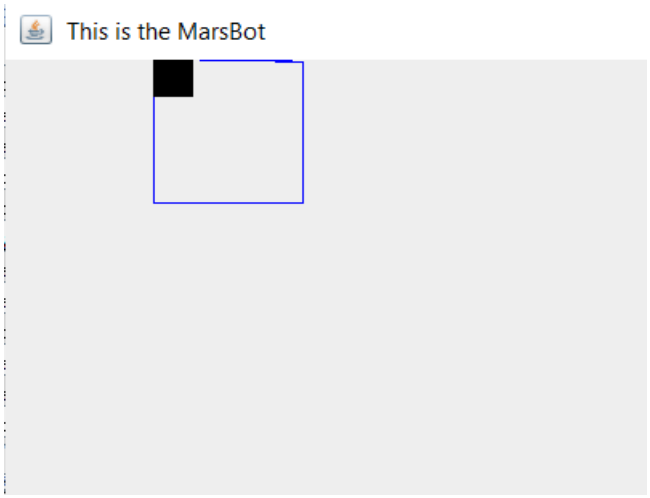
jal ROTATE

```
jal GO
sleep0:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
jal STOP

jal TRACK # Danh dau TRACK
addi $a0, $0, 180 # Quay goc
jal ROTATE
jal GO
sleep2:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau TRACK
addi $a0, $0, 90 # Quay goc
jal ROTATE
jal GO
sleep3:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu
```

```
jal TRACK # Danh dau TRACK
addi $a0, $0, 0 # Quay goc
jal ROTATE
jal GO
sleep4:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 2980
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau dia diem hien tai
addi $a0, $0, 270 # Quay sang trai de bat dau chay
jal ROTATE
jal GO
sleep1:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 3000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal STOP
endmain:
li $v0, 10
syscall

GO:
li $at, MOVING # Thay doi cong MOVING
```

```
addi $k0, $0, 1 # logic 1
sb $k0, ($at) # Bat dau chay
jr $ra
STOP:
li $at, MOVING # Thay doi cong MOVING
sb $0, ($at) # Dung chay
jr $ra
TRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
addi $k0, $0, 1 # logic 1
sb $k0, 0($at) # bat dau tracking
jr $ra
UNTRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
sb $0, 0($at) # dung ve
jr $ra
ROTATE:
li $at, HEADING # Thay doi cong HEAD
sw $a0, ($at) # Xoay robot
jr $ra
```

- Kết quả:




- **Vẽ hình sao:**

- Code:

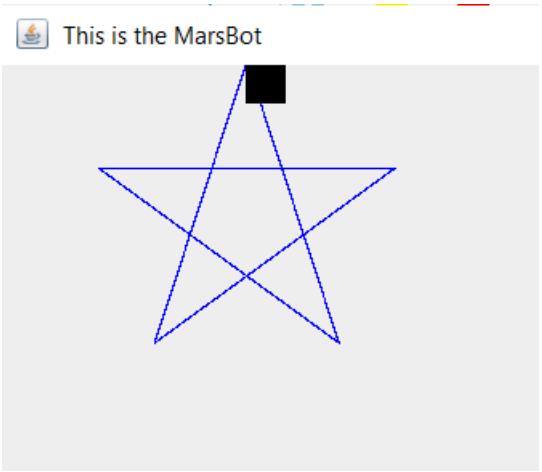
```
.eqv HEADING 0xffff8010 # Integer: Goc quay tu 0 den 359
# 0 : Tren
# 90: Phai
# 180: Duoi
# 270: Trai
.eqv MOVING 0xffff8050 # Boolean: Co di chuyen hay khong
.eqv LEAVETRACK 0xffff8020 # Boolean (0 hoac !0):
# Co track hay khong
.eqv WHEREX 0xffff8030 # Integer: Doc gia tri X hien tai cua con bot
.eqv WHEREY 0xffff8040 # Integer: Doc gia tri Y hien tai cua con bot
.text
main:
addi $a0, $0, 90 # Quay sang trai de bat dau chay
jal ROTATE
```

```
jal GO
sleep0:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 5000
syscall
jal STOP

jal TRACK # Danh dau dia diem hien tai
addi $a0, $0, 162 # Quay sang trai de bat dau chay
jal ROTATE
jal GO
sleep1:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 6000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau TRACK
addi $a0, $0, 306 # Quay goc
jal ROTATE
sleep2:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0,6000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau TRACK
```

```
addi $a0, $0, 90 # Quay goc
jal ROTATE
sleep3:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 6000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal TRACK # Danh dau TRACK
addi $a0, $0, 234 # Quay goc
jal ROTATE
sleep4:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 6000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu
jal TRACK # Danh dau TRACK
addi $a0, $0, 18 # Quay goc
jal ROTATE
sleep5:
addi $v0, $zero, 32 # De no chay trong 3000ms
li $a0, 6000
syscall
jal UNTRACK # Ve duong thang tu diem hien tai toi TRACK cu

jal STOP
endmain:
```

```
li $v0, 10
syscall

GO:
li $at, MOVING # Thay doi cong MOVING
addi $k0, $0, 1 # logic 1
sb $k0, ($at) # Bat dau chay
jr $ra
STOP:
li $at, MOVING # Thay doi cong MOVING
sb $0, ($at) # Dung chay
jr $ra
TRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
addi $k0, $0, 1 # logic 1
sb $k0, 0($at) # bat dau tracking
jr $ra
UNTRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
sb $0, 0($at) # dung ve
jr $ra
ROTATE:
li $at, HEADING # Thay doi cong HEAD
sw $a0, ($at) # Xoay robot
jr $ra
```

- Kết quả:




## Assignment 2

- **Code:**

```
.eqv KEY_CODE 0xffff0004 # ASCII tu ban phim, 1 byte
.eqv KEY_READY 0xffff0000 # =1 neu ma co ky tu moi
# tu dong clear sau lw
.eqv DISPLAY_CODE 0xffff000c # ASCII de show, 1 byte
.eqv DISPLAY_READY 0xffff0008 # = 1 neu ma san sang ghi, clear sau sw
.text
li $k0, KEY_CODE
li $k1, KEY_READY
li $s0, DISPLAY_CODE
li $s1, DISPLAY_READY
li $s2, 64
li $s3, 91
li $s4, 96
li $s5, 123
```

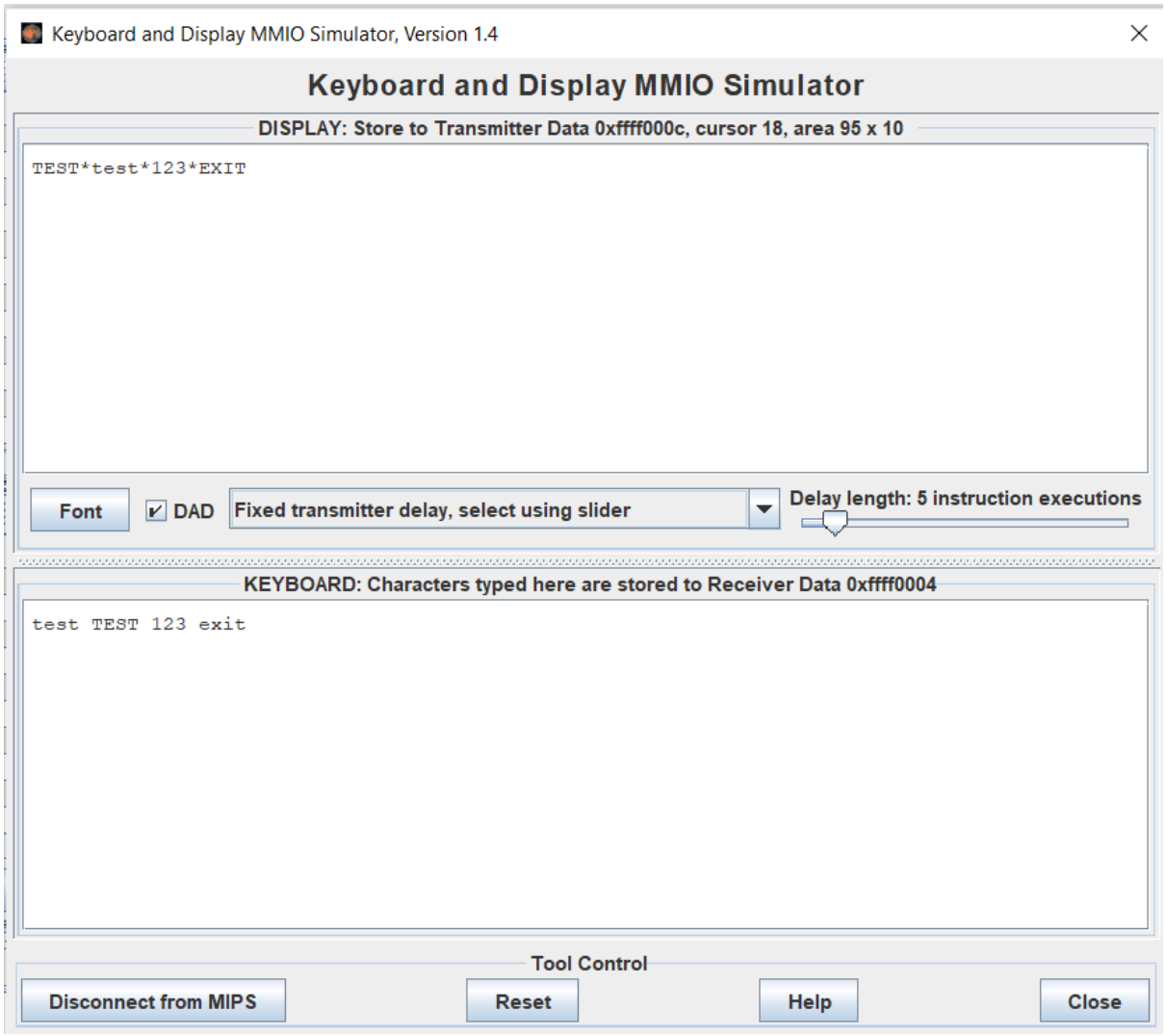
```
li $t5, 1
li $t6, 2
li $t7, 3
li $t8, 47
li $t9, 58
li $s7, 0
loop:
nop
WaitForKey:
lw $t1, ($k1) # t1 = [k1] = KEY_READY
beq $t1, $0, WaitForKey # key == 0 => wait

ReadKey:
lw $t0, ($k0) # Doc ky tu

WaitForDis:
lw $t2, ($s1)
beq $t2, $0, WaitForDis

ToLower:
sgt $t2, $t0, $s2 # if t0 >= 65 && t0 <= 90
slt $t3, $t0, $s3
and $t4, $t2, $t3
beqz $t4, ToUpper # If true => toupper
addi $t0, $t0, 32
j ShowKey
```

```
ToUpper:
sgt $t2, $t0, $s4 # if t0 >= 96 && t0 <= 123
slt $t3, $t0, $s5
and $t4, $t2, $t3
beqz $t4, Number # If true => number
subi $t0, $t0, 32
j ShowKey

Number:
sgt $t2, $t0, $t8 # if t0 >= 47 && t0 <= 58
slt $t3, $t0, $t9
and $t4, $t2, $t3
beqz $t4, Null # If true => do nothing
j ShowKey

Null:
addi $t0, $0, 42

ShowKey:
sw $t0, ($s0)
addi $t1, $0, 69 # ky tu e
addi $t2, $0, 88 # ky tu x
addi $t3, $0, 73 # ky tu i
addi $t4, $0, 84 # ky tu t
beq $s7, $0, put_1
beq $s7, $t5, put_2
beq $s7, $t6, put_3
```

```
beq $s7, $t7, put_4

next:
lw $t0, 12($sp)
seq $t1, $t0, $t1
lw $t0, 8($sp)
seq $t2, $t0, $t2
lw $t0, 4($sp)
seq $t3, $t0, $t3
lw $t0, 0($sp)
seq $t4, $t0, $t4
and $t1, $t1, $t2
and $t1, $t1, $t3
and $t1, $t1, $t4
bnez $t1, exit
j loop

put_1:
addi $s7, $s7, 1
lw $s6, 12($sp)
beq $s6, $t1, put_2
add $s7, $0, $0
sw $t0, 12($sp)
addi $s6, $s6, 1
j next
put_2:
addi $s7, $s7, 1
```

```
lw $s6, 8($sp)
beq $s6, $t2, put_3
add $s7, $0, $0
sw $t0, 8($sp)
addi $s6, $s6, 1
j next
put_3:
addi $s7, $s7, 1
lw $s6, 4($sp)
beq $s6, $t3, put_4
add $s7, $0, $0
sw $t0, 4($sp)
addi $s6, $s6, 1
j next
put_4:
add $s7, $0, $0
lw $s6, 0($sp)
beq $s6, $t4, exit
sw $t0, 0($sp)
add $s6, $0, $0
j next

exit:
li $v0, 10
syscall
```

- Kết quả:




## Assignment 3

- **Code:**

```
.eqv KEY_CODE 0xffff0004 # ASCII tu ban phim, 1 byte
.eqv KEY_READY 0xffff0000 # =1 neu ma co ky tu moi
# tu dong clear sau lw
.eqv DISPLAY_CODE 0xffff000c # ASCII de show, 1 byte
```

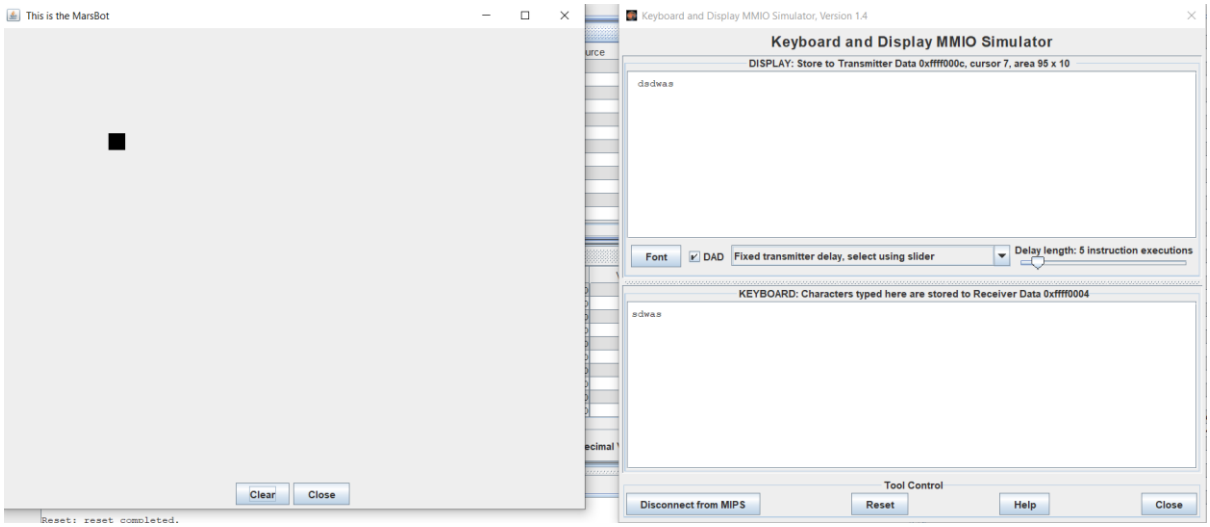
```
.eqv DISPLAY_READY 0xffff0008 # = 1 neu ma san sang ghi, clear sau sw
.eqv HEADING 0xffff8010 # Integer: Goc quay tu 0 den 359
# 0 : Tren
# 90: Phai
# 180: Duoi
# 270: Trai
.eqv MOVING 0xffff8050 # Boolean: Co di chuyen hay khong
.eqv LEAVETRACK 0xffff8020 # Boolean (0 hoac !0):
# Co track hay khong
.eqv WHEREX 0xffff8030 # Integer: Doc gia tri X hien tai cua con bot
.eqv WHEREY 0xffff8040 # Integer: Doc gia tri Y hien tai cua con bot
.text
li $a2, KEY_CODE
li $a3, KEY_READY
li $s0, DISPLAY_CODE
li $s1, DISPLAY_READY
li $t8, 0 # Check xem co dang chay hay khong
li $t3, 32 # dau cach
li $t4, 119 # w
li $t5, 97 # a
li $t6, 115 # s
li $t7, 100 # d

loop:
nop
WaitForKey:
lw $t1, ($a3) # t1 = [k1] = KEY_READY
```

```
beq $t1, $0, WaitForKey # key == 0 => wait

ReadKey:
lw $t0, 0($a2) # Doc ky tu

WaitForDis:
lw $t2, ($s1)
beq $t2, $0, WaitForDis

ShowKey:
sw $t0, ($s0)
beq $t0, $t4, UP
beq $t0, $t5, LEFT
beq $t0, $t6, DOWN
beq $t0, $t7, RIGHT
beq $t0, $t3, RUN_STOP # space

next:
nop
j loop

UP:
addi $a0, $0, 0
jal ROTATE
j next

LEFT:
```

```
addi $a0, $0, 270
jal ROTATE
j next

DOWN:
addi $a0, $0, 180
jal ROTATE
j next

RIGHT:
addi $a0, $0, 90
jal ROTATE
j next

RUN_STOP:
beq $t8, $0, RUN
jal STOP
addi $t8, $0, 0
j next
RUN:
jal GO
addi $a0, $0, 90
jal ROTATE
addi $t8, $0, 1
j next
GO:
li $at, MOVING # Thay doi cong MOVING
```

```
addi $k0, $0, 1 # logic 1
sb $k0, ($at) # Bat dau chay
jr $ra
STOP:
li $at, MOVING # Thay doi cong MOVING
sb $0, ($at) # Dung chay
jr $ra
TRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
addi $k0, $0, 1 # logic 1
sb $k0, 0($at) # bat dau tracking
jr $ra
UNTRACK:
li $at, LEAVETRACK # Thay doi cong LEAVETRACK
sb $0, 0($at) # dung ve
jr $ra
ROTATE:
li $at, HEADING # Thay doi cong HEAD
sw $a0, ($at) # Xoay robot
jr $ra
```

- **Kết quả**


This is the MarsBot
Clear
Close
Reset: reset completed.
Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c, cursor 7, area 95 x 10
dadwas
Font
DAD
Fixed transmitter delay, select using slider
Delay length: 5 instruction executions
KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004
sdwas
Tool Control
Disconnect from MIPS
Reset
Help
Close