Curriculum Development for a Weekend Robotics Experience Series

Chi N. Thai

www.cntrobotics.com\iteea-2018

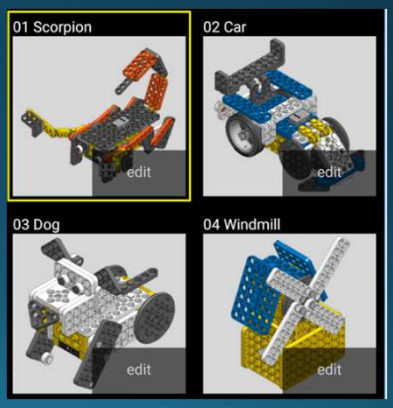
WRE Series @ UGA's GA Center

- Middle School Students.
- Core Session (2 days): D1 & D2
  - *Foundational Robotics Knowledge using ROBOTIS PLAY700 kit.*
  - *Autonomous Behavior & Remote Control.*
- Follow-up Session A (1 day): D3
  - *Multimedia Programming using mobile devices.*
- Follow-up Session B (1 day): D4
  - *Robotics Programming with SCRATCH 2.*
  - *Multimedia Enhancement using SCRATCH 2.*

ROBOTIS PLAY 700

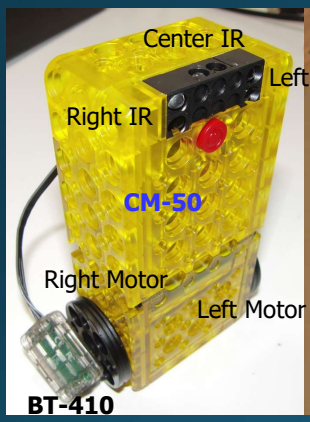
(Courtesy of ROBOTIS, Inc.)

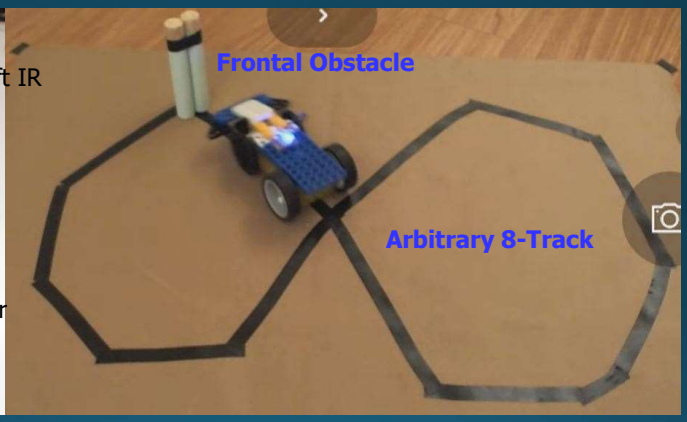
$110
For USA only



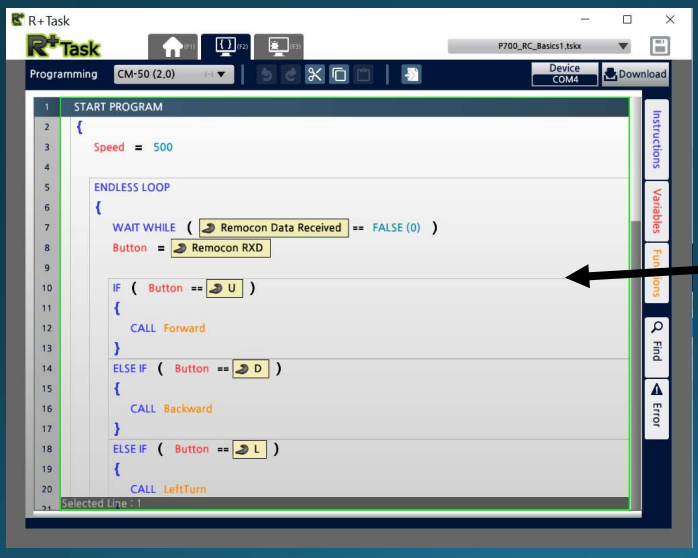
PLAY 700 Plate/Rivet System

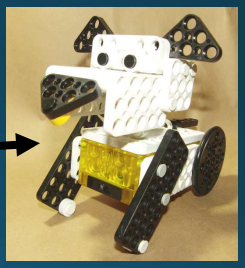
PLAY 700 Sensors & Actuators





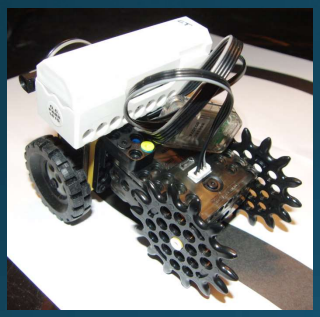
CM-50 & TASK v.2 (Windows/iOS/Android)





3 ways to program/operate robot (1)

- Robot operates all by itself using "TASK" tool:
  - Autonomous Behavior and Remote Control.



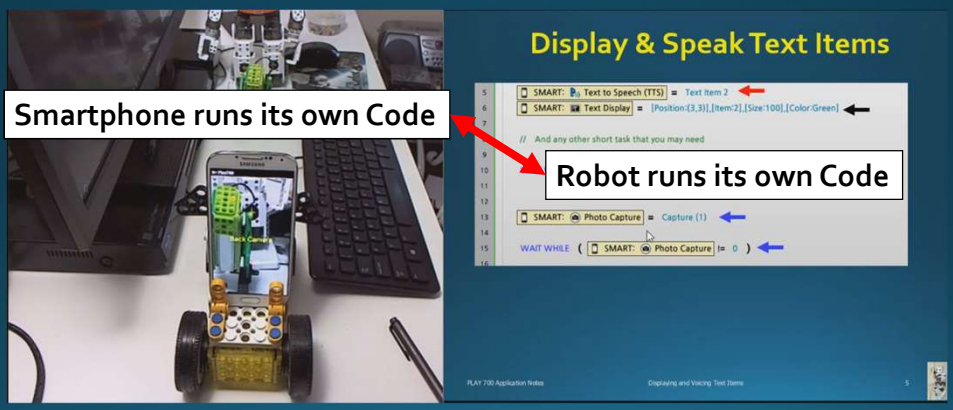
```
// ROBOTIS Dream Level 2 : Avoider Sample Program
// Obstacle Avoider

START PROGRAM
{
  // Plays a melody when turned on.
  Buzzer Timer = Melody Time
  Buzzer Index = Melody No.3 (3)
  WAIT WHILE ( Buzzer Timer > 0.0sec )

  Speed = 400
  LineDistance = 500
  ObjectDistance = 100

  ENDLESS LOOP
  {
    Print Screen = IR Left
    Print Screen = IR Center
    Print Screen with Line = IR Right
```

Computer Code runs on Robot

Core Session

3 ways to program/operate robot (2)

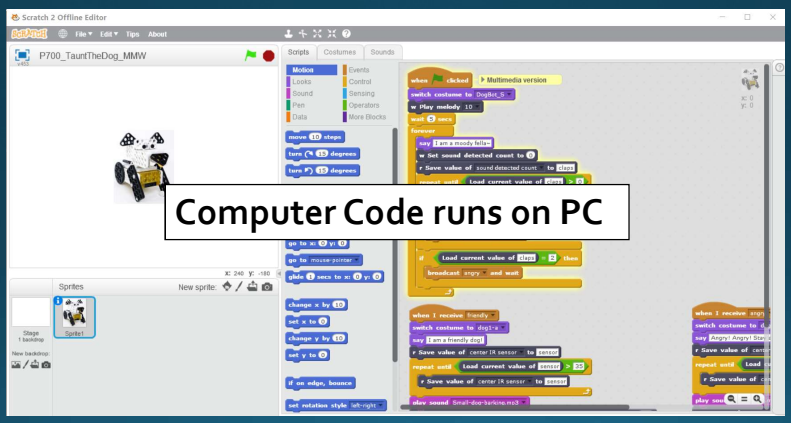
- Collaborative Behavior (with another computer):
  - Using Smart Phone's tools.



Follow-up Session A

3 ways to program/operate robot (3)

- Supervisory Control of "Dumb" Robot:
  - using SCRATCH 2.



Follow-up Session B

Curriculum Development Issues

- Students have diverse backgrounds and skill levels (manual & technical):
  - *Core materials (structured, scaffolding & streamlined).*
  - *Optional materials (depending on students & events on actual training day).*
  - *Student Motivation Level (mostly High).*

- "Spiral" Approach:
  - *Pedagogical Patterns: Advice For Educators (Bergin – 2012).*

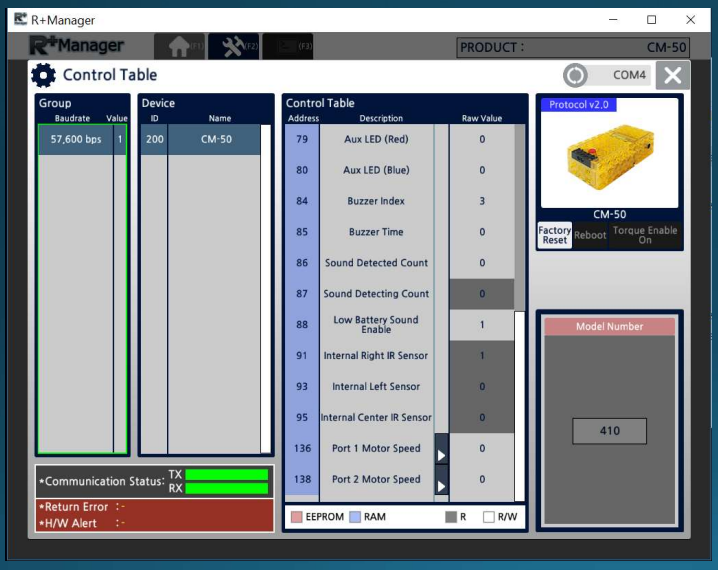
D1-AM

Foundational Knowledge (0)

Using ROBOTIS Tools

D1-AM

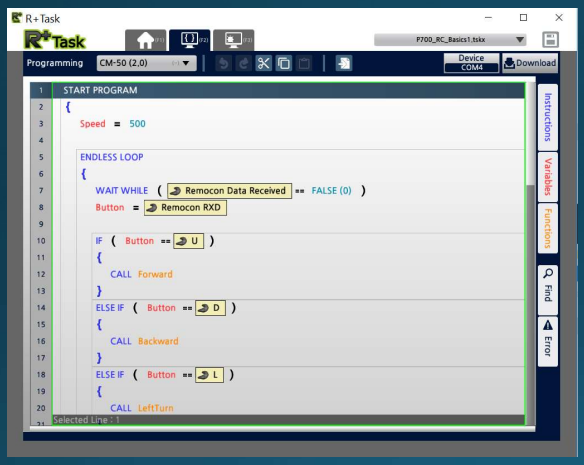
MANAGER V.2 & CM-50

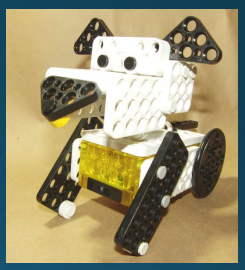


- LEDs
- Motors (1 or 2)
- NIR Sensors (LCR)
- Sound Detector
- Music Generator
- **Firmware Update**

D1-AM

CM-50 & TASK v.2 (Mobile & MS Windows)





D1-AM

R+TASK Practice on "Spinning Top"

- Print & Println procedures. (Sequence of Actions)
- How to read/print NIR Sensors. (Endless Loop)
- How to use Hi-Res Timer. (Wait While Loop)
- How to turn on/off Geared Motors. (Wait While Loop)
- Use NIR Sensor Values for Motor Speed Control. (Combining Conditional & While Loop structures)
- Robot Maneuvers Programming. (Program Modularization using Function)

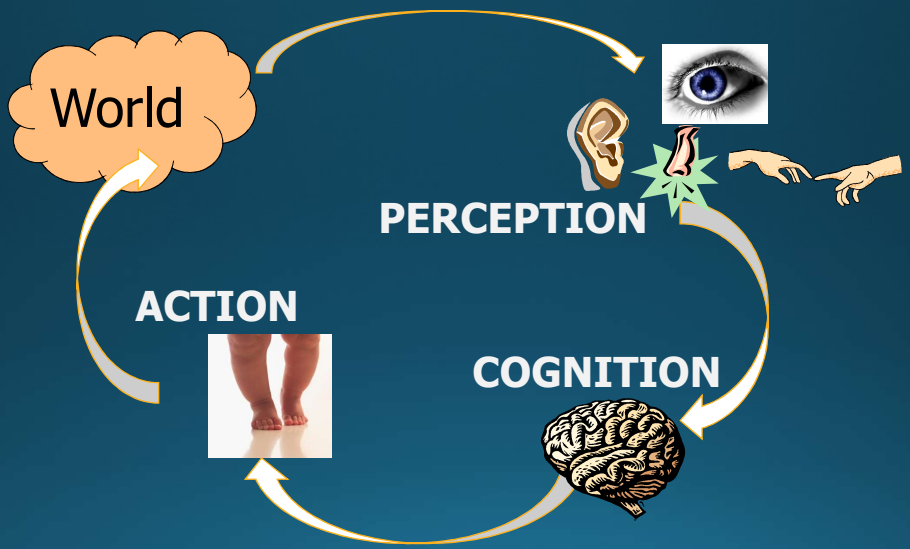
Foundational Knowledge (1)

D1-D2

Autonomous Behavior

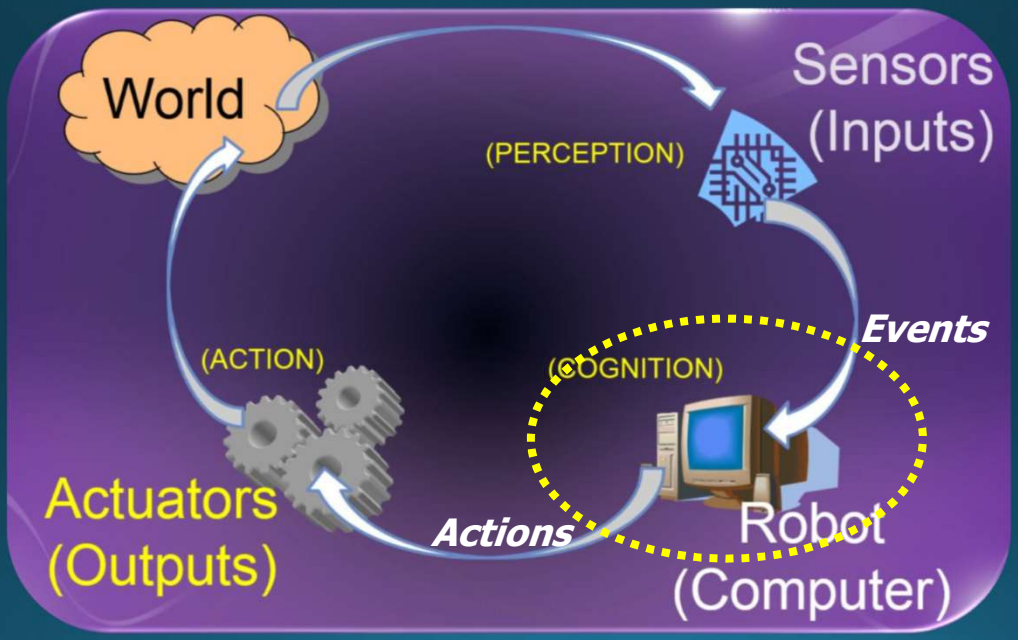
World – Human Interactions

D1-AM



World - Robot Interactions

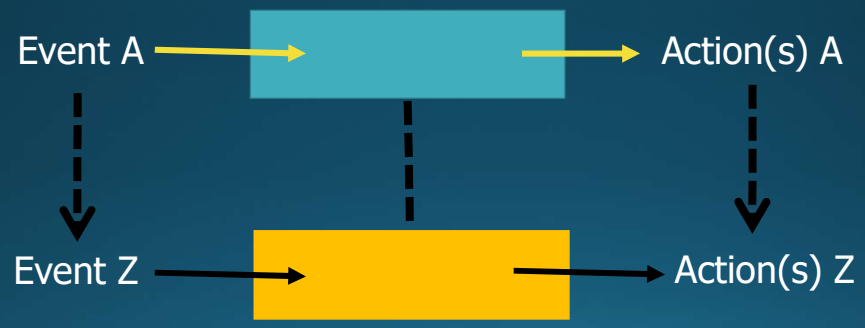
D1-AM



Event Processing Approach (1)

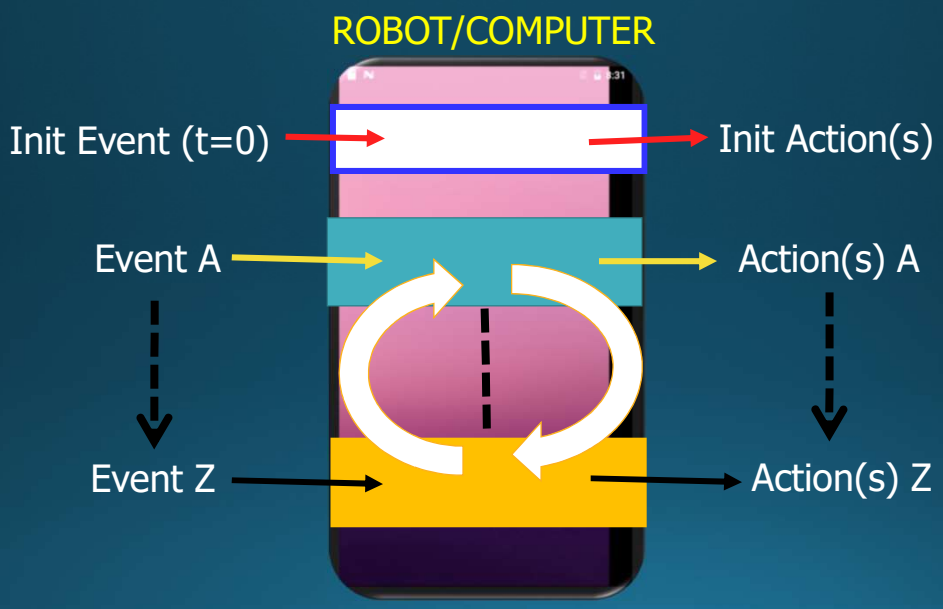
D1-AM

THINK of SELF as ROBOT



D1-AM

Event Processing Approach (2)



D1-AM

Reactive Control Approach (1)

Given Condition (Event) >> Appropriate Robot Action

Events/Conditions	Actions
Condition 1	Action A
Condition 2	Action B
Condition 3	Action C
Condition 4	Action B

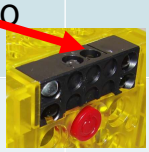
- One and Only One Condition happens at any one time
- Multiple Conditions can happen at any one time

D1-AM

Reactive Control Approach (2)

Given Condition (Event) >> Appropriate Robot Action

Sensor Check	Actuator Operation
IR Center Sensor < 50	Bot Stays Still
IR Center [50-300}	Bot Spins Right
IR Center [300-500}	Bot Spins Left
IR Center > 500	Bot Fast Forward



1 and Only 1 Condition (Event) can happen at any 1 time

Using a new Programming Structure: Parallel IFs

D1-AM

Parallel IFs

```
4   // Init Action
5   Speed = 256
```

MAPPING Condition & Action items into specific TASK code items

Think ANALOGIES

Note Bot's runtime performance

```
8   ENDLESS LOOP
9   {
10    IR_Sensor = IR Center
11    IF ( IR_Sensor < 50 )
12    {
13      CALL Stop
14    }
15    IF ( IR_Sensor >= 50 && IR_Sensor < 300 )
16    {
17      CALL Right
18    }
19    IF ( IR_Sensor >= 300 && IR_Sensor < 500 )
20    {
21      CALL Left
22    }
23    IF ( IR_Sensor > 500 )
24    {
25      Speed = 512
26      CALL Forward
27    }
28  }
```

D1-AM

IF-ELSE-IF

```
4   // Init Action
5   Speed = 256
```

More Efficient Computationally with IF-ELSE-IF

Speed proportional to IR Sensor reading

```
8   ENDLESS LOOP
9   {
10    IR_Sensor = IR Center
11    Speed = IR_Sensor
12    IF ( IR_Sensor < 50 )
13    {
14      CALL Stop
15    }
16    ELSE IF ( IR_Sensor < 300 )
17    {
18      CALL Right
19    }
20    ELSE IF ( IR_Sensor < 500 )
21    {
22      CALL Left
23    }
24    ELSE
25    {
26      CALL Forward
27    }
28  }
```


D1-AM

COMPARING RUN-TIME PERFORMANCES

```
13  High-resolution Timer = 1.000sec
```

```
43  TimeElapsed = 1000 - High-resolution Timer
44  Print Screen with Line = TimeElapsed
```

```
27  IF ( IR_Sensor >= 300 && IR_Sensor <= 500 )
28  {
29    Speed = IR_Sensor
30    CALL Left
31    Buzzer Timer = 0.1sec
32    Buzzer Index = Re# (6)
33    WAIT WHILE ( Buzzer Timer > 0.0sec )
34  }
```

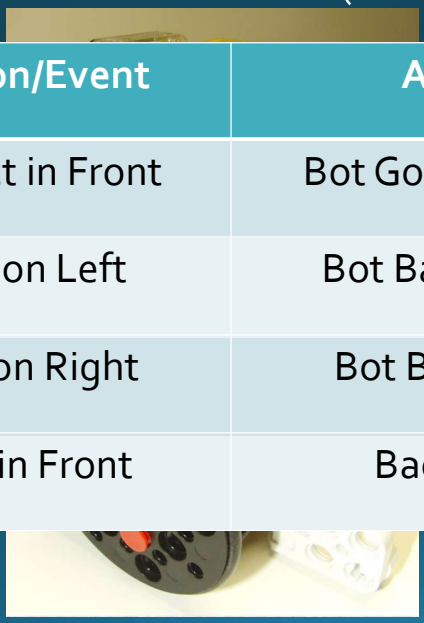
```
27  ELSE IF ( IR_Sensor <= 500 )
28  {
29    Speed = IR_Sensor
30    CALL Left
31    Buzzer Timer = 0.1sec
32    Buzzer Index = Re# (6)
33    WAIT WHILE ( Buzzer Timer > 0.0sec )
34  }
```

D1-PM

"Avoider" Robot

AVOIDER uses 2 NIR Sensors (Left & Right).

Condition/Event	Action
No Object in Front	Bot Goes Forward
Object on Left	Bot Backs Right
Object on Right	Bot Backs Left
Object in Front	Backward



D1-PM

IF-ELSE-IF solution

Avoider Robot

```
IR_Left  =  IR Left
IR_Right =  IR Right
IF ( IR_Left < 10 && IR_Right < 10 )
{
   CALL Forward
}
ELSE IF ( IR_Left >= 100 && IR_Right < 10 )
{
   CALL RightBack
}
ELSE IF ( IR_Right >= 100 && IR_Left < 10 )
{
   CALL LeftBack
}
ELSE IF ( IR_Left >= 100 && IR_Right >= 100 )
{
   CALL Backward
}
```

D1-PM

"Avoider" Refinement (1)

Re-Analysis of Condition/Action Table:

"Object in Front" behavior needs modifying.

Condition/Event	Action
No Object in Front	Bot Goes Forward
Object on Left	Bot Backs Right
Object on Right	Bot Backs Left
Object in Front	Backward + LeftBack

D1-PM

"Avoider" Refinement (2)

Re-Analysis of Condition/Action Table:

Find DEFAULT BEHAVIOR (Going Forward).

Find ESSENTIAL BEHAVIORS (Backs Right and Backs Left).

Condition/Event	Action
No Object in Front	Bot Goes Forward
Object on Left	Bot Backs Right
Object on Right	Bot Backs Left
~~Object in Front~~	~~Backward + LeftBack~~

D1-PM

Parallel IFs revisited

```
IR_Left  =  IR Left
IR_Right =  IR Right
IF ( IR_Left >= 100 )
{
   CALL RightBack
   CALL Delay
}
IF ( IR_Right >= 100 )
{
   CALL LeftBack
   CALL Delay
}
CALL Forward
```

RE-ANALYSIS of CONDITION/ACTION List

ESSENTIAL BEHAVIORS

Give Robot TIME to do its job (2)

DEFAULT BEHAVIOR

D1-AM (earlier)

"Give Robot TIME to do its job" (1)

```
START PROGRAM
{
   Speed = 500
   Print Screen = Speed
   Print Screen with Line = 1111
   Print Screen = 0000 0000 0000 0000 0000 0000 1111 0011
}
```

```
   PORT[1]:Geared Motor = CCW:512 (50.05%)
}
```

DOES NOT WORK!

```
   PORT[1]:Geared Motor = CCW:512 (50.05%)
   High-resolution Timer = 1.000sec
   WAIT WHILE ( High-resolution Timer > 0.000sec )
}
```

Avoider - Parallel LOOPs (1)

D1-PM?

Optional when Time permits

```
12  IR_Left  =  IR Left
13  IR_Right  =  IR Right
14  LOOP WHILE  (  IR_Left  >=  100  )
15  {
16     CALL  RightBack
17
18
19  LOOP WHILE  (  IR_Right  >=  100  )
20  {
21     CALL  LeftBack
22     CALL  Delay
23  }
24  CALL  Forward
```

DOES NOT WORK!

Avoider - Parallel LOOPs (2)

D1-PM?

LOOPs need "fresh" data

NESTED LOOPs

DELAYs Not needed

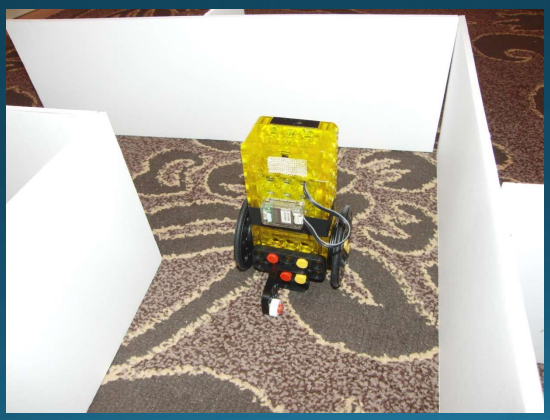
```
10  ENDLESS LOOP
11  {
12     //  IR_Left  =  IR Left
13     //  IR_Right  =  IR Right
14     LOOP WHILE  (  IR Left  >=  100  )
15     {
16        CALL  RightBack
17        CALL  Delay
18     }
19     LOOP WHILE  (  IR Right  >=  100  )
20     {
21        CALL  LeftBack
22        CALL  Delay
23     }
24     CALL  Forward
```

D1-PM

"Emergent" Code Behaviors

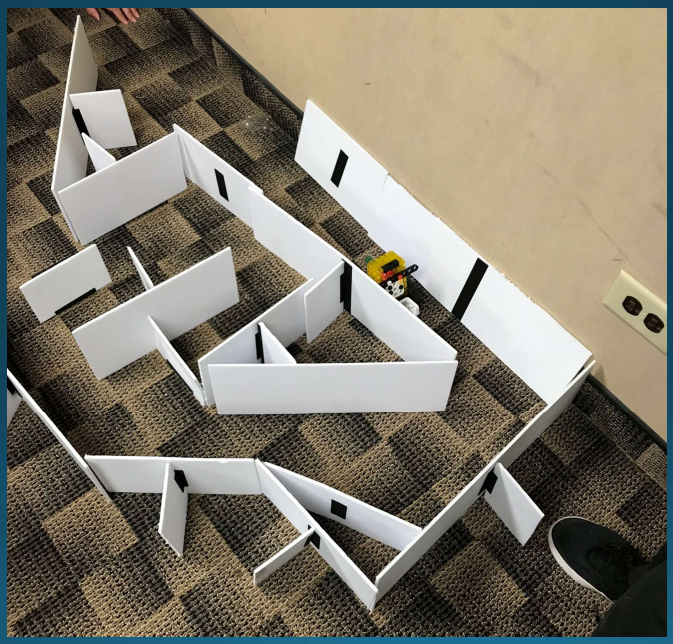
Try various AVOIDER versions to see how Robot behaves in dealing with a Hallway.

Why would certain codes work or not work?



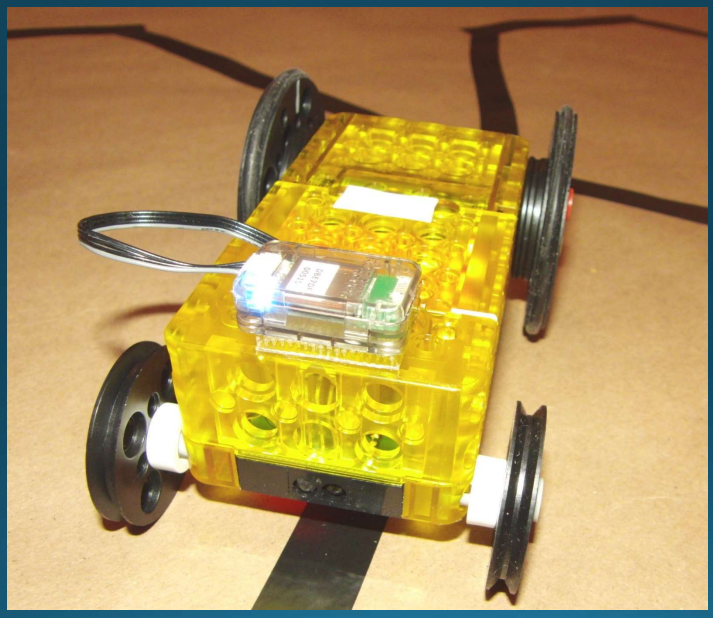
D1-PM

Student Version of Hallway/Maze



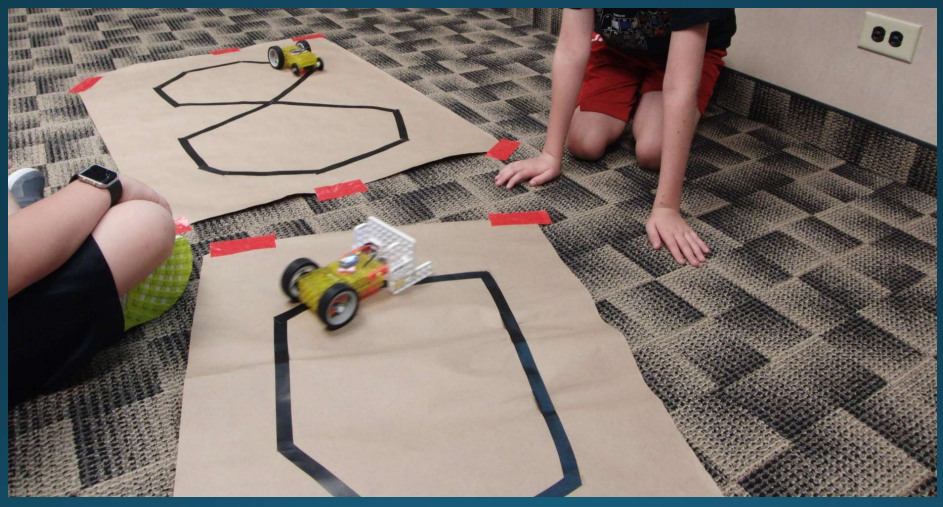
D2-PM

Simple CarBot



D2-PM

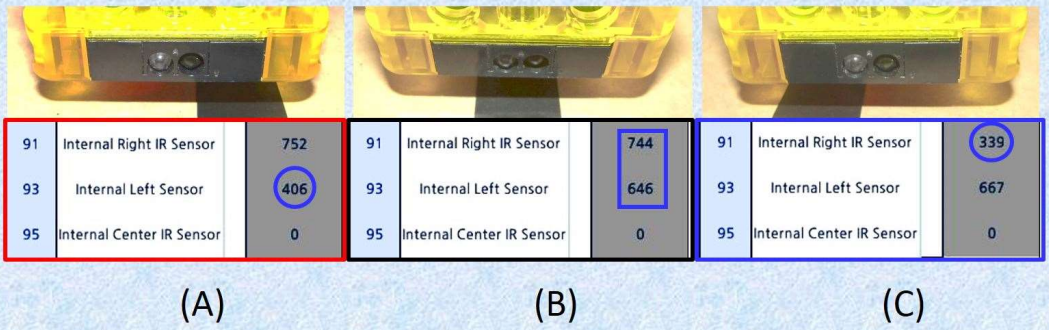
Line Follower Project



Analysis of L & R IR Sensors Behaviors

D2-PM

IR Sensors Readings of Black Track



(A) (B) (C)

CONDITION/ACTION solution

D2-PM

Reactive Control Approach

Given Condition (Event) >> Appropriate Robot Action

Condition/Event	Action
Car on top of Track	Car Goes Forward
Car left of Track	Car Turns Right
Car right of Track	Car Turns Left

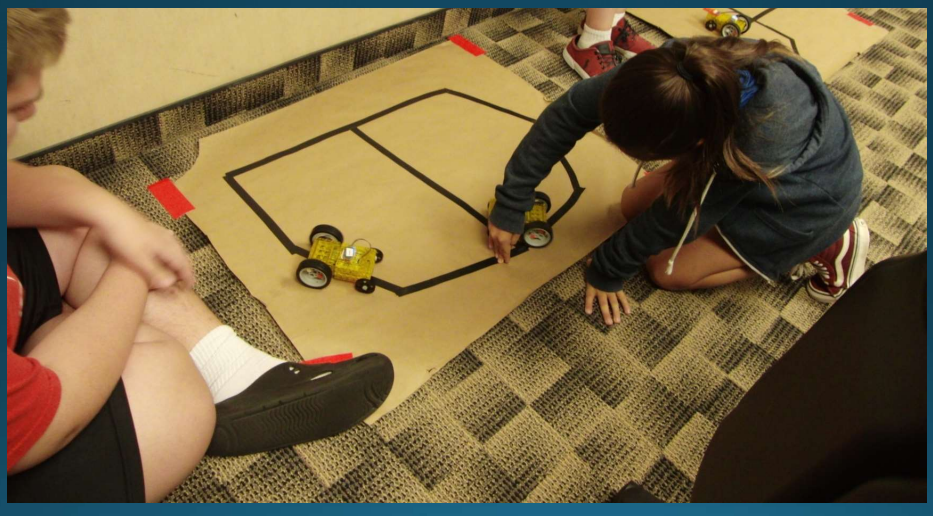
D2-PM

TASK solution

```
IF ( IR Left < LineDistance && IR Right >= LineDistance )
{
  // Turns left when ONLY the left sensor detects black.
  CALL LeftTurn
}
ELSE IF ( IR Left >= LineDistance && IR Right < LineDistance )
{
  // Turns right when ONLY the right sensor detects black
  CALL RightTurn
}
ELSE
{
  // Moves forward for all other cases.
  CALL Forward
}
```


D2-PM

LineFollower_ObstacleAvoider (Complex Conditional Maneuver)



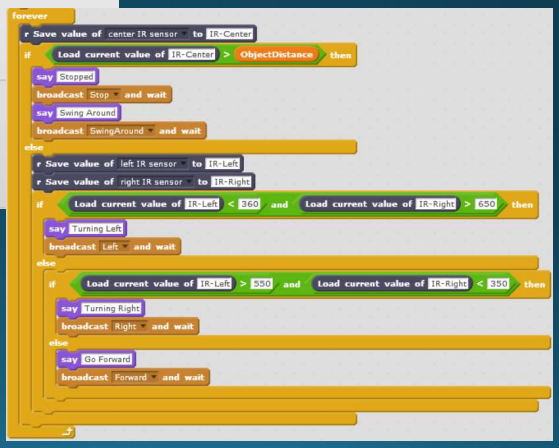
D4-AM

TASK vs. SCRATCH 2

```
IF ( IR Left < LineDistance && IR Right >= LineDistance )
{
  // Turns left when ONLY the left sensor detects black.
  CALL LeftTurn
}
ELSE IF ( IR Left >= LineDistance && IR Right < LineDistance )
{
  // Turns right when ONLY the right sensor detects black
  CALL RightTurn
}
ELSE
{
  // Moves forward for all other cases.
  CALL Forward
}
```

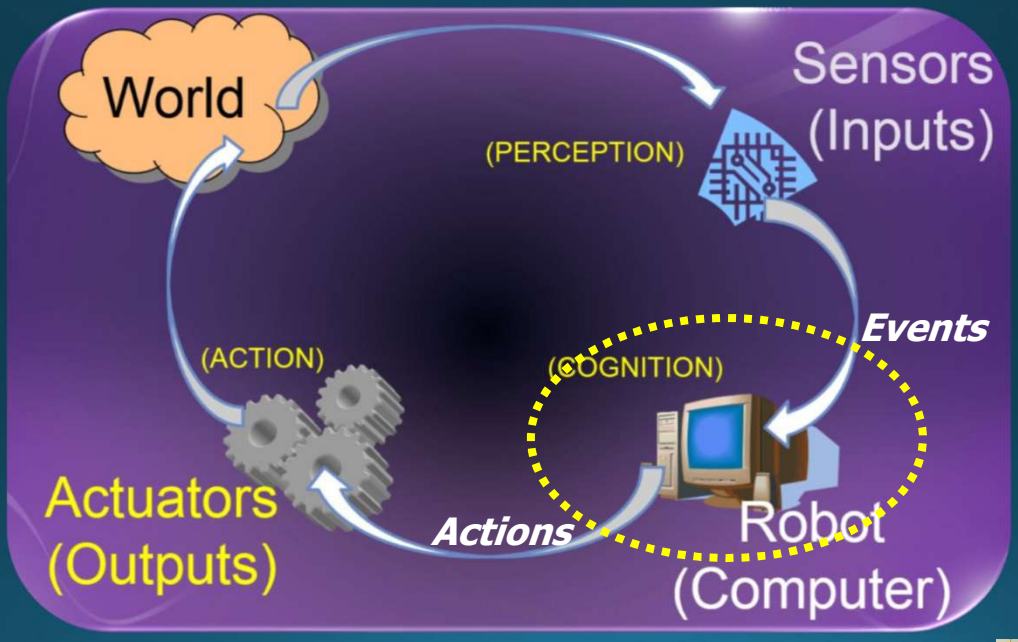
Core Session

Follow-up Session B



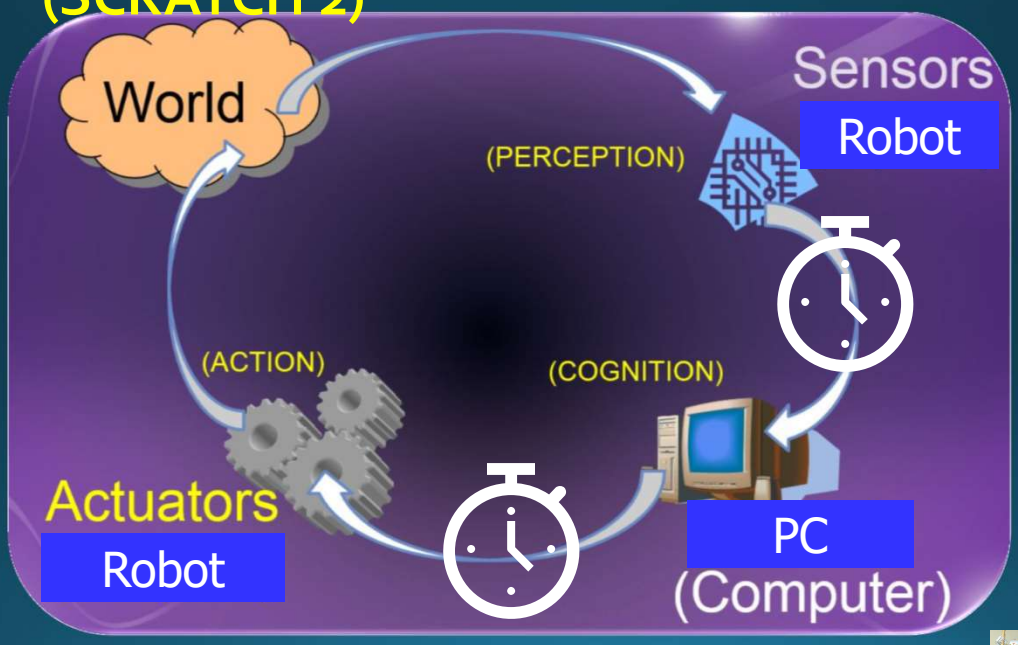
D4-AM

World - Robot Interactions



World - Robot Interactions (SCRATCH 2)

D4-AM



ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 43

Foundational Knowledge (2)

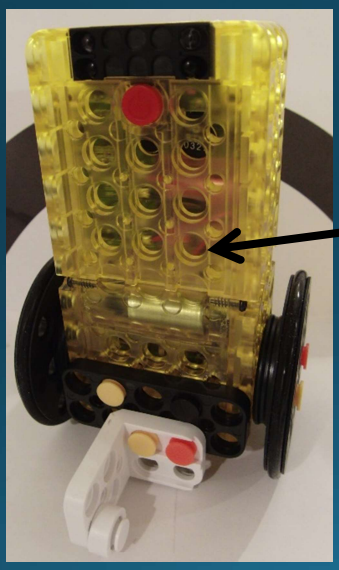
D1-D2

Remote Control

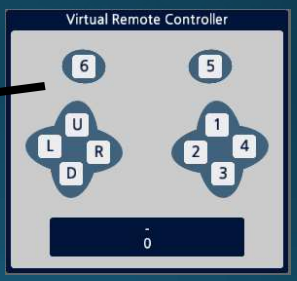
ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 44

RC of Spinning Top Robot

D1-PM



RC-100 Virtual Controller from PC



D1-PM

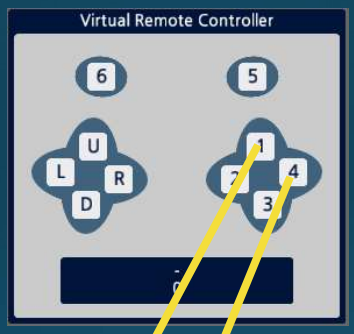
RC_Basics1.tskx (1)

```
1  START PROGRAM
2  {
3      Speed = 500
4
5      ENDLESS LOOP
6      {
7          WAIT WHILE ( Remocon Data Received == FALSE (0) )
8          Button = Remocon RXD
9
10         IF ( Button == U )
11         {
12             CALL Forward
13         }
14         ELSE IF ( Button == D )
15         {
16             CALL Backward
17         }
```

IF-ELSE-IF for only 1 condition at a time

D1-PM

RC Programming Challenge 1



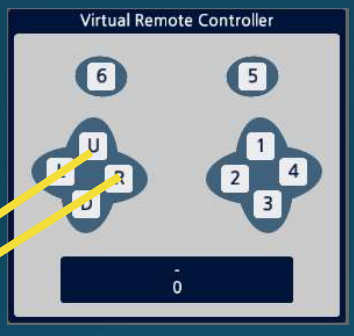

```
31      ELSE IF ( ? == ? )
32      {
33
34      }
```

BUTTON 1 >> Low Speed (Speed = 300)
BUTTON 4 >> High Speed (Speed = 700)

Start coding from "P_RCBasics2_Challenge1.tskx"

D1-PM

RC Programming Challenge 2



"U+R" >> Robot goes Forward then Right,
i.e. a Wider/Gentler Right Turn.

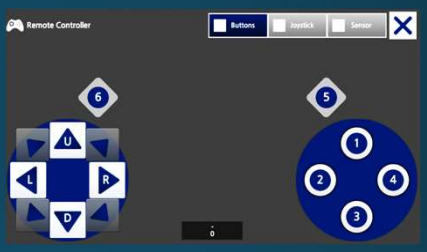
Create new "P_RC-Basics2Challenge2.tskx"

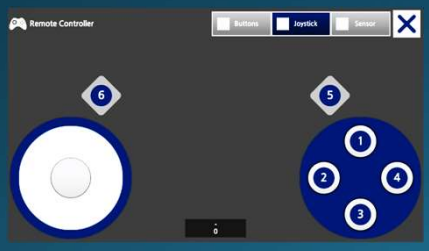
Give Robot TIME to do its job (3)

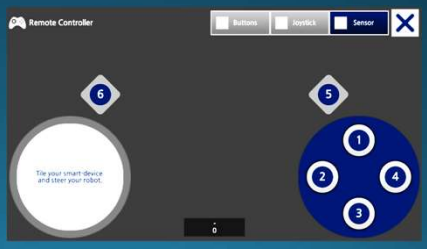
D1-PM

"Mobile" Virtual Controller

- Use the virtual controller from TASK tool on Mobile Device.
- In preparation for SumoBot competition.

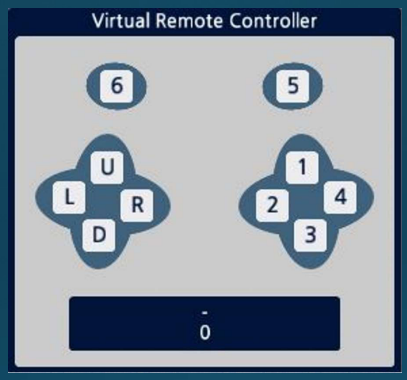






D2-AM

RC with Multiple Buttons



RC-100 10-bit Message

Bit Position	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Button	**6**	**5**	**4**	**3**	**2**	**1**	**R**	**L**	**D**	**U**
Example Data	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**

Masking Operation to access only Wanted Bits (1)

D2-AM

AND Operation with Mask U-D-L-R

Bit Position	9	8	7	6	5	4	3	2	1	0
Button	6	5	4	3	2	1	R	L	D	U
Example Data	0	0	0	1	0	0	0	1	0	0
							&			
Mask U+D+L+R	0	0	0	0	0	0	1	1	1	1
Result from & Op.	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*

A	B	A & B
1	0	0
1	1	1
0	0	0
0	1	0

Masking Operation to access only Wanted Bits (2)

D2-AM

AND Operation with Mask 1-2-3-4

Bit Position	9	8	7	6	5	4	3	2	1	0
Button	6	5	4	3	2	1	R	L	D	U
Example Data	0	0	0	1	0	0	0	1	0	0
						&				
Mask 1+2+3+4	0	0	1	1	1	1	0	0	0	0
Result from & Op.	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*0*

A	B	A & B
1	0	0
1	1	1
0	0	0
0	1	0

RC-MultiDirectionsSpeeds.tskx (1)

D2-AM

```
20  ENDLESS LOOP
21  {
22      WAIT WHILE ( Remocon Data Received == FALSE (0) )
23      Data = Remocon RXD
24      Print Screen with Line = Data
25      IF ( Data == -- )
26      {
27          CALL Stop
28      }
29      ELSE
30      {
31          // Filtering out specific U-D-L-R-1-3 buttons
32          ButtonUp = Data & U
33          ButtonDown = Data & D
34          ButtonLeft = Data & L
35          ButtonRight = Data & R
36          Button1 = Data & 1
37          Button3 = Data & 3
```


RC-MultiDirectionsSpeeds.tskx (2)

D2-AM

```
38  IF ( ButtonUp == U )
39  {
40      CALL Forward
41  }
42  IF ( ButtonDown == D )
43  {
44      CALL Backward
45  }
46  IF ( ButtonLeft == L )
47  {
48      CALL Left
49  }
50  IF ( ButtonRight == R )
51  {
52      CALL Right
53  }
54  IF ( Button1 == 1 )
55  {
56      Speed = 200
57  }
58  IF ( Button3 == 3 )
59  {
60      Speed = 300
61  }
```

Foundational Knowledge (3)

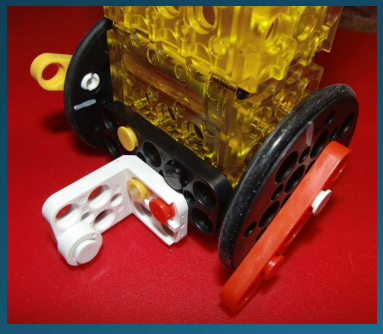
D1-D2

Mechanical Engineering

Wheel Designs for CarBot

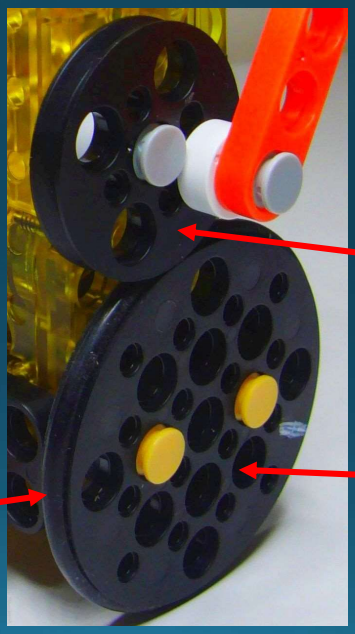
D1-PM (last)






Gear-like Functionality

- The PLAY700 kit does not have actual GEAR components, but ones can obtain "gear-like" functionality using the "pulley-wheels" and the "rubber tires" that came with the kit.

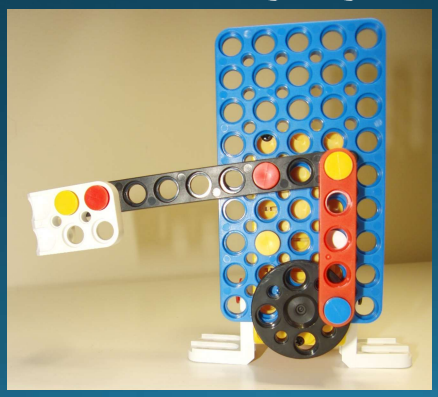


Student 4-wheel drive design

D2-PM (demos)

4-Bar Linkage Web Sites

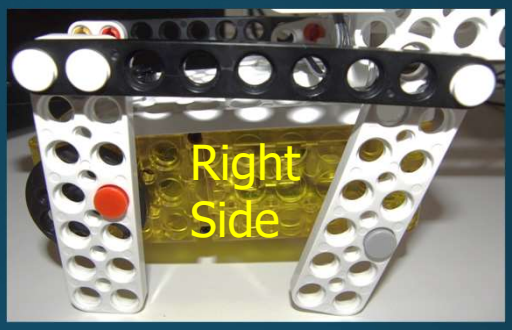
- Wikipedia entry (fixed, input, coupler, follower links)
- 4-Bar Animation (http://www.mekanizmalar.com/fourbar.html)
  - Try different linkage configurations and note effects on walking motion.
- KMODDL (Kinematic Models for Design Digital Library) – Cornell University.

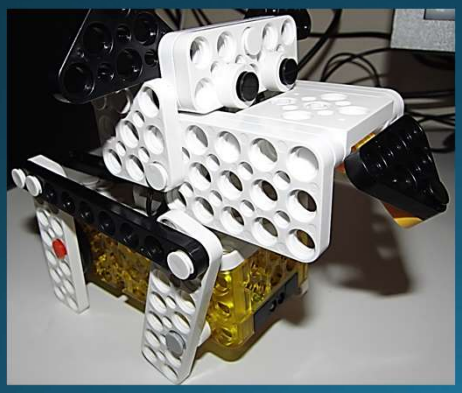


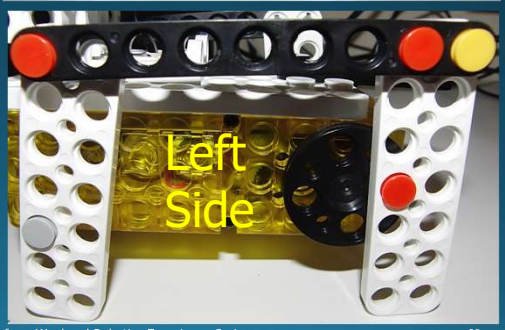
ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 59

D2-PM (demos)

Walking 4-legged DogBot





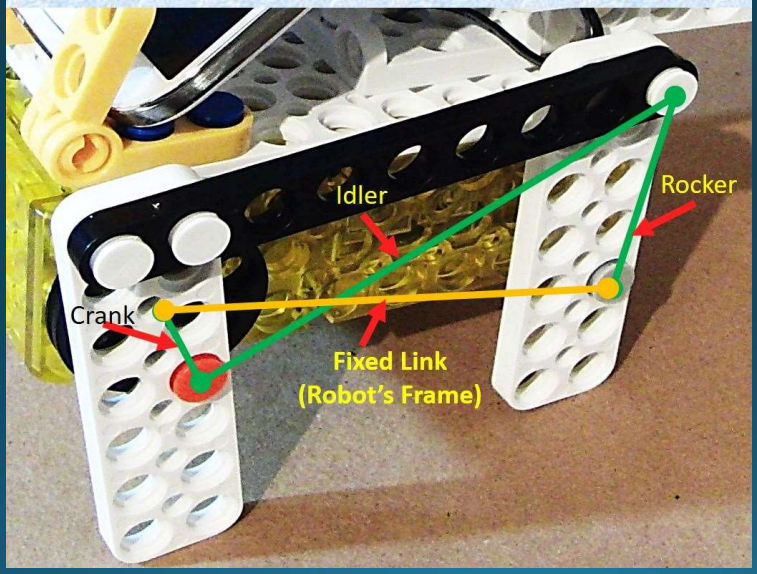


ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 60

D2-PM (demos)

4-Bar Linkage Concept

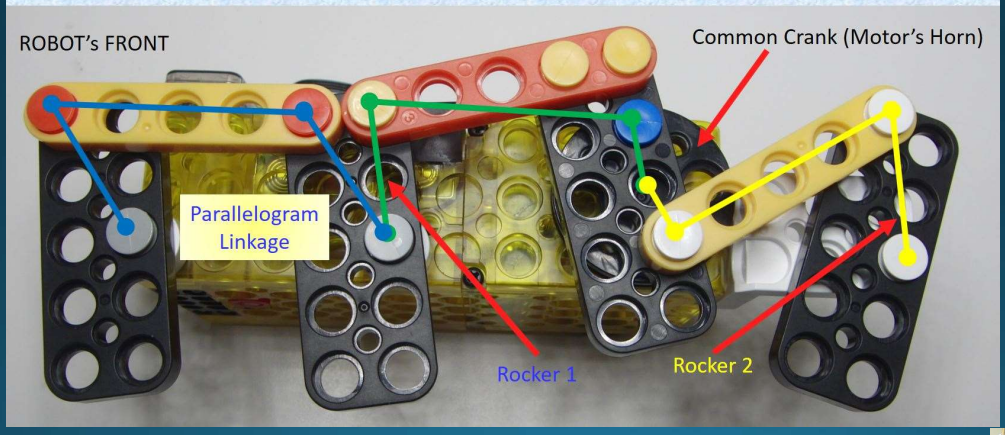
DogBot Legs



D2-PM (demos)

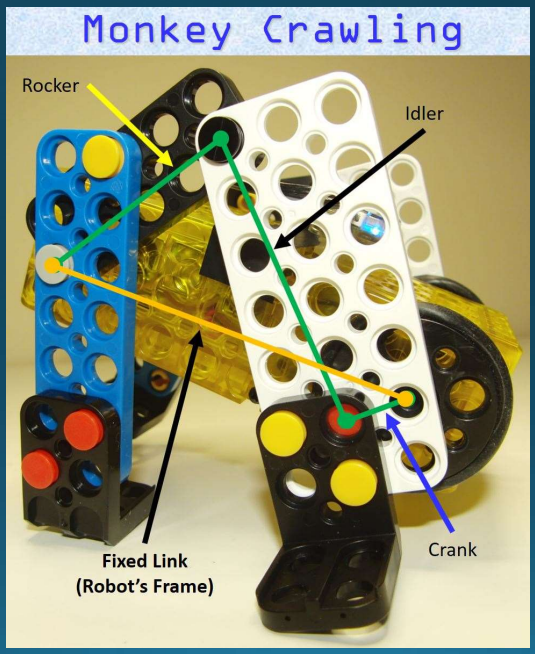
Scorpion Robot

Scorpion Legs



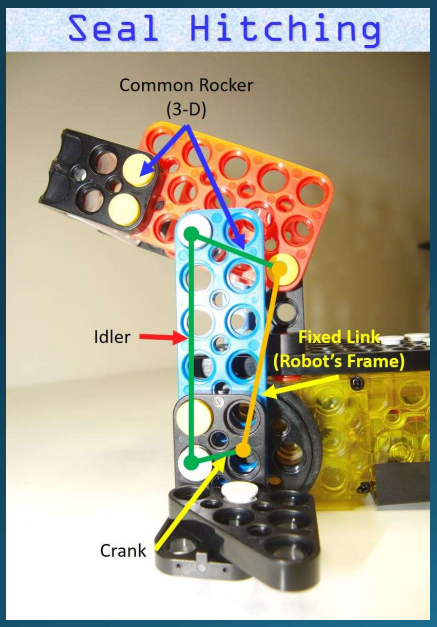
D2-PM (demos)

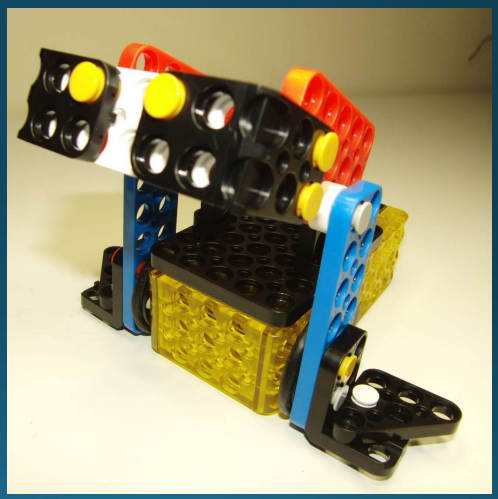
Monkey



D2-PM (demos)

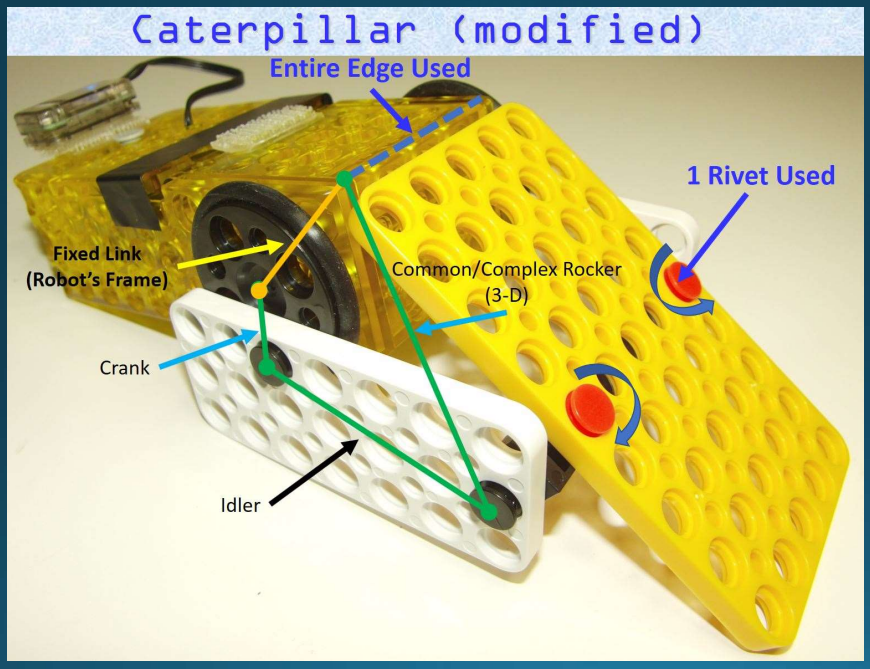
Seal (synchronized L-R linkages)





4-Bar Linkage on Caterpillar

D2-PM (demos)



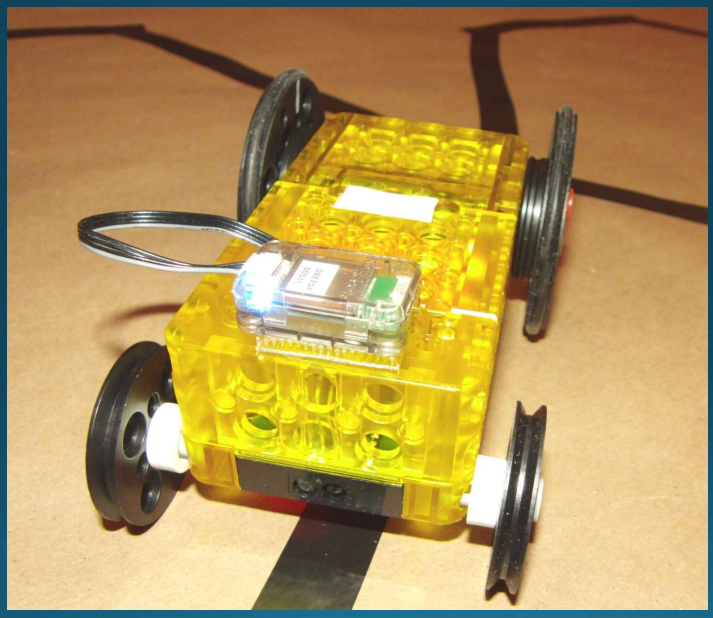
Foundational Knowledge (4)

D1-D2

Integration Practice

Simple CarBot or Build Your Own CarBot (At Home)

D1-PM

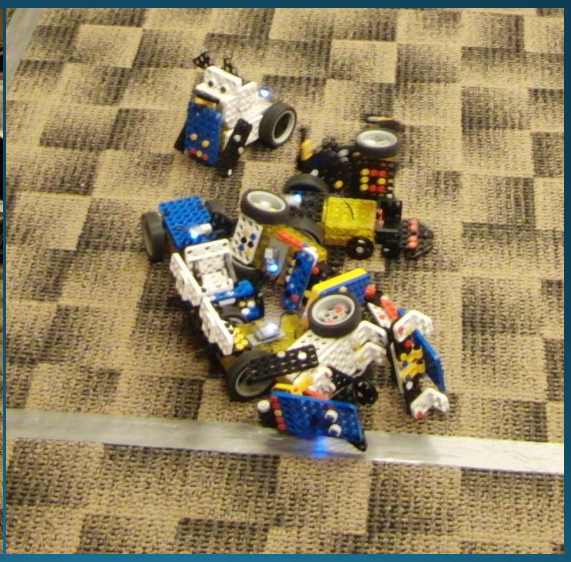


Finish Building Competition Bot

D2-PM

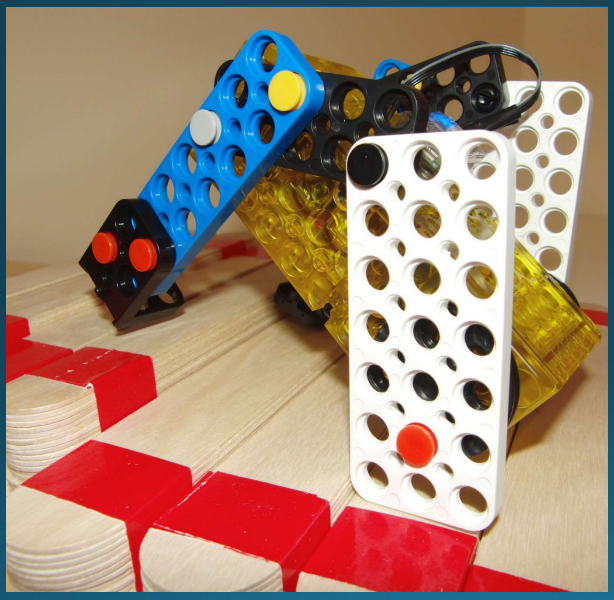
D2-PM

Sumo Bot Competition



D2-PM

Mountain Climbing Competition (planned)



D4

Foundational Knowledge (5)

Supervisory Control (via SCRATCH 2)

D4-AM

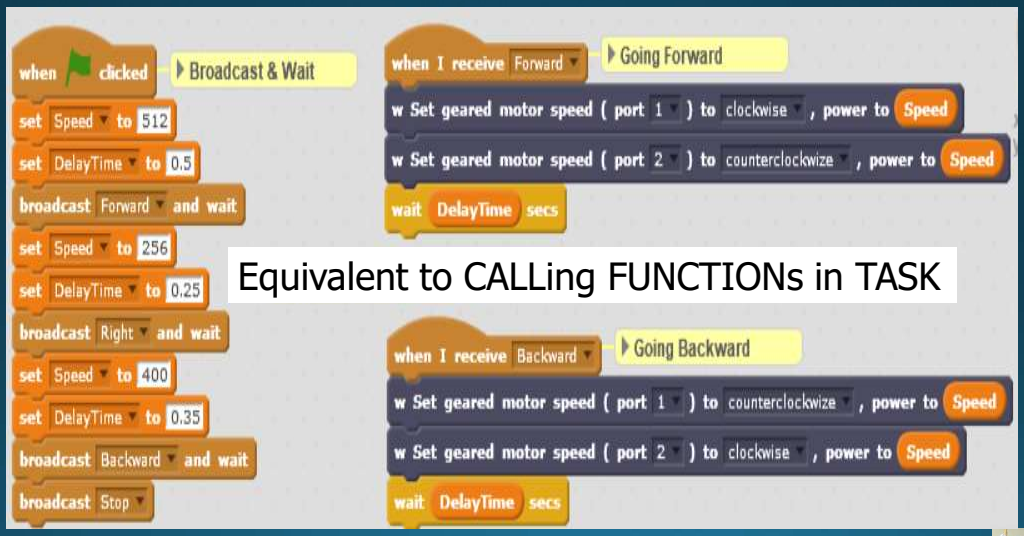
SCRATCH 2: Event Processing (1)

Keyboard Processing

D4-AM

SCRATCH 2: Event Processing (2)

Broadcast & Wait



Equivalent to CALLing FUNCTIONs in TASK

D4-AM

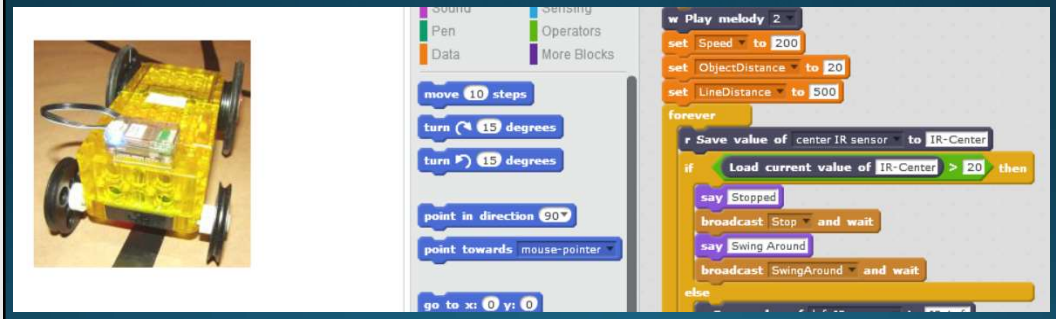
SCRATCH 2: Event Processing (3)

Broadcast
DOES NOT WORK for Robotics

D4-PM

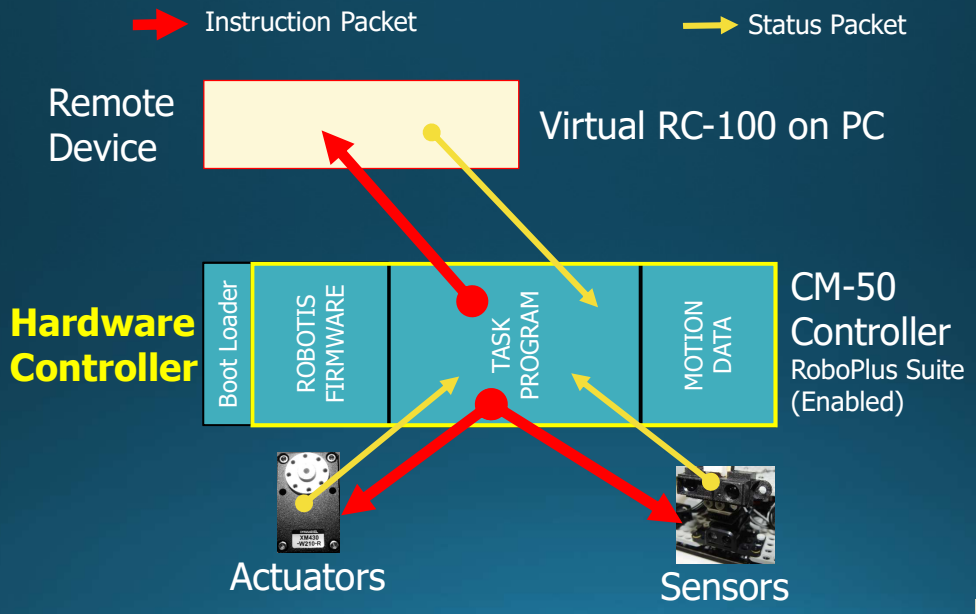
SCRATCH 2: Event Processing (4)

Communications Throughput
Not Fast Enough for
Demanding Autonomous Behaviors



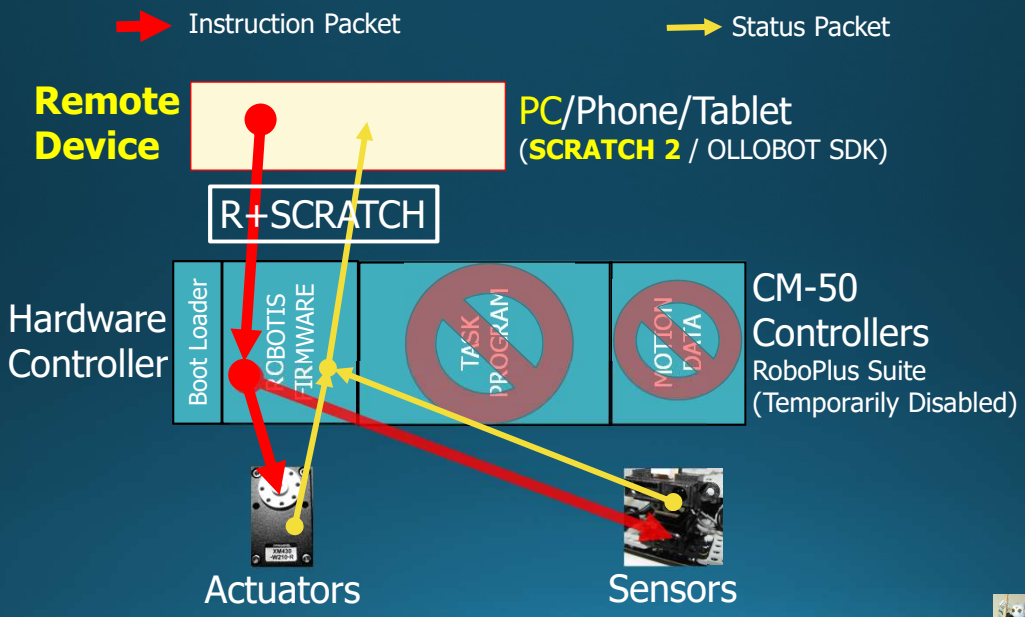
D4-PM

Communications Hierarchy (TASK)



Communications Hierarchy (SCRATCH 2)

D4-PM



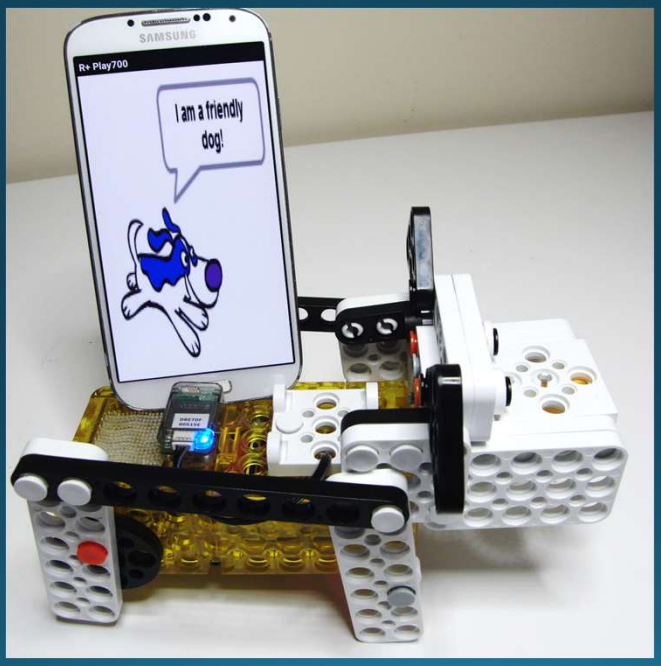
Foundational Knowledge (6)

D3

Collaborative Control (via Mobile Device)

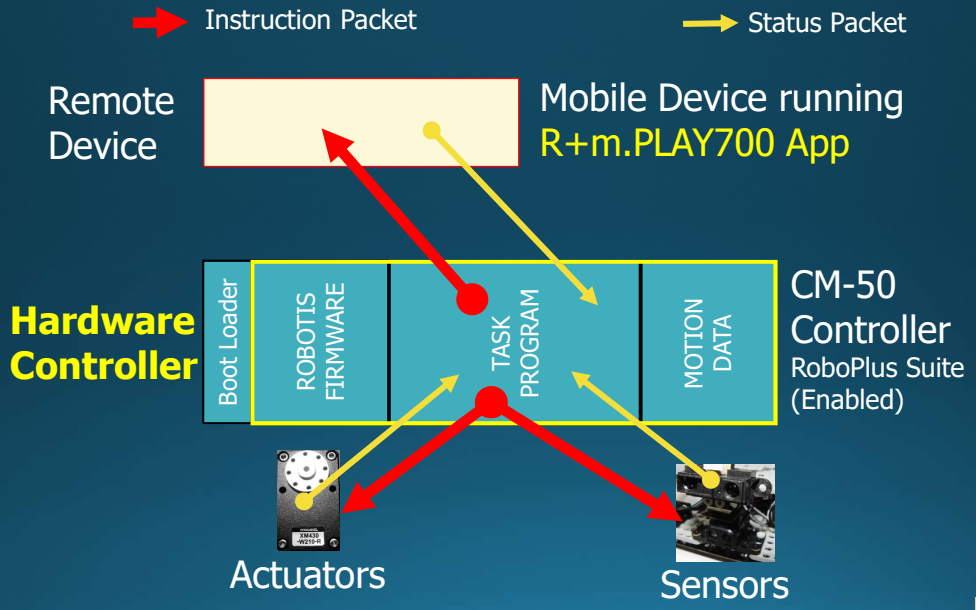
R+m.PLAY700 App

D3-AM



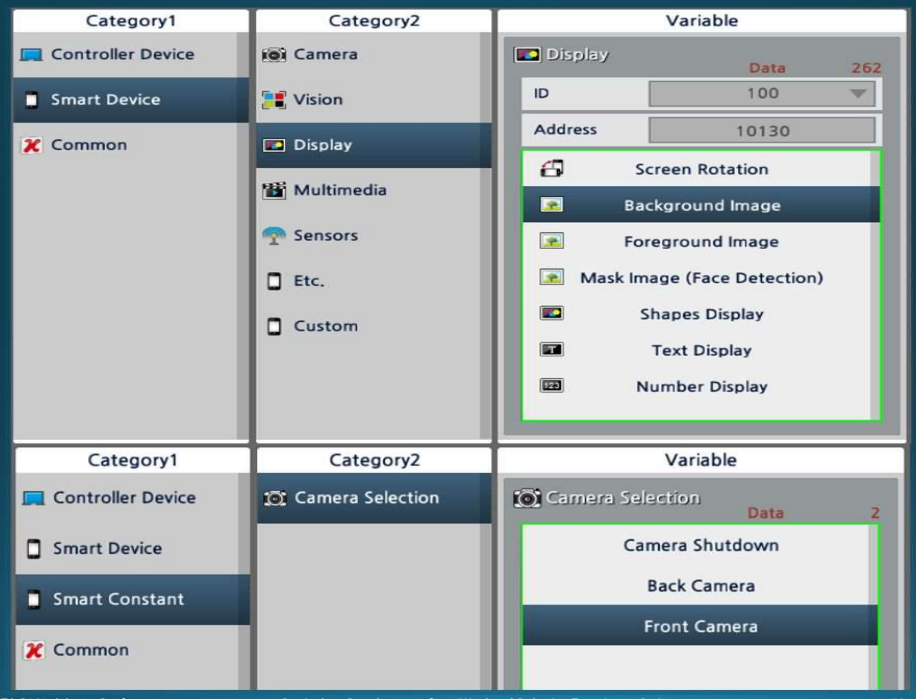
Communications Hierarchy (TASK/R+m.PLAY700)

D3-AM



"SMART" Commands in TASK (1)

D3-AM



"SMART" Commands in TASK (2)

D3-AM

Text/Graphics & Touch Areas Applications


```
ENDLESS LOOP
{
  CALL BAKCGROUND_MUSIC
  TOUCH = SMART: Touch Area 1
  IF ( TOUCH == 17 )
  {
    SMART: Vibration Time = 10
    CALL BACKWARD
  }
  ELSE IF ( TOUCH == 22 )
  {
    SMART: Vibration Time = 10
    PORT[1]:Geared Motor = CW:0 (0.00%) + 300
    PORT[2]:Geared Motor = CCW:0 (0.00%) + 300
  }
  ELSE IF ( TOUCH == 21 )
  {
    SMART: Vibration Time = 10
    CALL TURN_LEFT
```

D3-PM

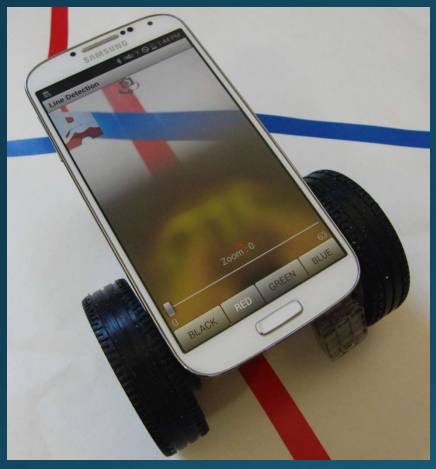
R+m.PLAY700 Speech Tool

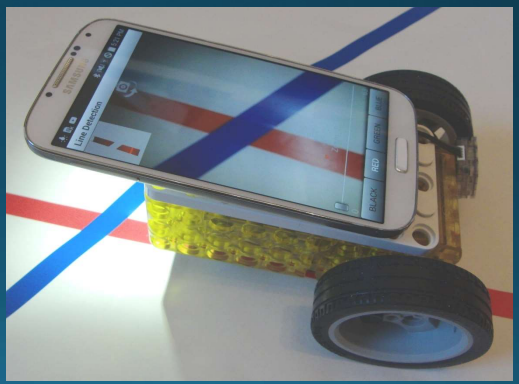
Text to Speech & Voice Recognition

Voice Recognition	Add
1	ready
2	forward
3	backward
4	turn right
5	turn left
6	stop
7	invalid command
8	valid commands

D3-PM

ColorLineFollower





D3-PM

Intruder Alert Application

```
14  ENDLESS LOOP
15  {
16    DetectedZone = SMART: Motion Detection Area
17    IF ( DetectedZone != FALSE (0) )
18    {
19      SMART: Photo Capture = Capture (1)
20      Buzzer Timer = Melody Time
21      Buzzer Index = Melody No.20 (20)
22      WAIT WHILE ( Buzzer Timer > 0.000sec )
23      WAIT WHILE ( SMART: Photo Capture != 0 )
24      SMART: Smart Timer = 1.0sec
25      WAIT WHILE ( SMART: Smart Timer > 0.000sec )
26    }
27  }
```


WRE 2018 Student Feedback

Feb 16-17

Robot Construction Skills

BEFORE/AFTER you take this Robotics training, how would you rate your skill level in robot construction:

No. of Students	BEFORE	AFTER
4	**Low**	**Medium**
2	**Medium**	**High**
1	**Low**	**High**
2	Medium	Medium
2	High	High

64%

ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 87

Mar 24-25

Robot Construction Skills

BEFORE/AFTER you take this Robotics training, how would you rate your skill level in robot construction:

No. of Students	BEFORE	AFTER
3	**Low**	**Medium**
3	**Medium**	**High**
2	Medium	Medium
1	*Medium*	*Low*

67%

ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 88

Feb 16-17

Robot Programming Skills

BEFORE/AFTER you take this Robotics training, how would you rate your skill level in robotics programming:

No. of Students	BEFORE	AFTER
3	**Low**	**Medium**
4	**Medium**	**High**
2	Medium	Medium
1	High	High
1	*High*	*Medium*

64%

ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 89

Mar 24-25

Robot Programming Skills

BEFORE/AFTER you take this Robotics training, how would you rate your skill level in robotics programming:

No. of Students	BEFORE	AFTER
3	**Low**	**Medium**
2	**Medium**	**High**
1	**Low**	**High**
1	**Medium**	**Very High**
1	*Low*	*Low*

87%

ITEEA 2018 Atlanta Conference Curriculum Development for a Weekend Robotics Experience Series 90