江西理工大學 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Mobile application development

移动应用开发

Lecture 07:

APP Inventor_environment_C

Dr Ata Jahangir Moshayedi

Prof Associate ,
School of information engineering Jiangxi
university of science and technology, China

EMAIL: ajm@jxust.edu.cn

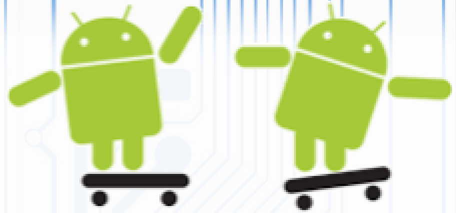
Autumn _2021

- From this lecture any question from teacher side which student didn’t answer consider as the student absent and will report to school

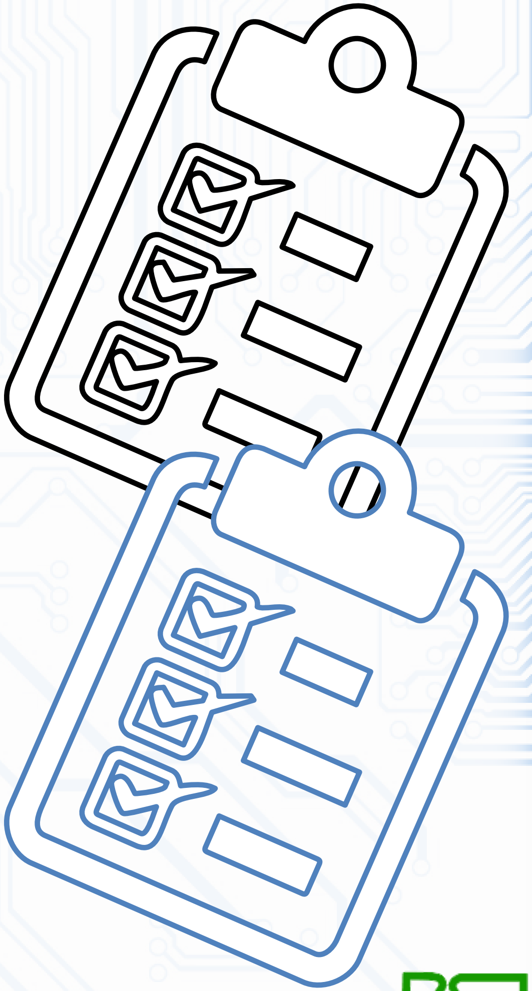
MOBILE APPLICATION DEVELOPMENT

LECTURE 08: APP Inventor_environment_C

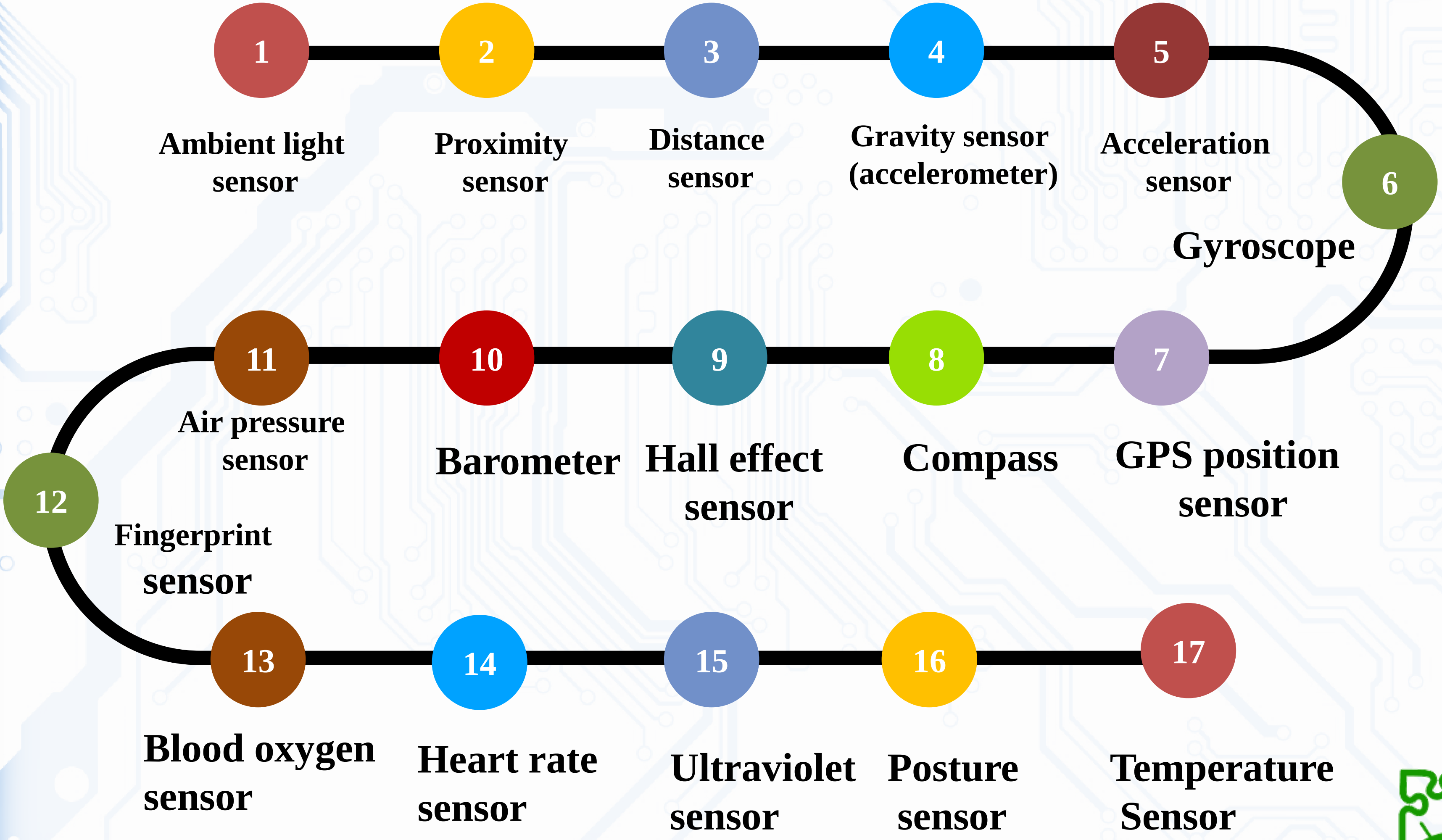
Agenda



- ➢Smartphone sensors
  - ✓Sensors
    - ✓Social
      - ✓Built-in blocks



Smartphone sensors



1 Ambient light sensor

2 Proximity sensor

3 Distance sensor

4 Gravity sensor (accelerometer)

5 Acceleration sensor

6 Gyroscope

7 GPS position sensor

8 Compass

9 Hall effect sensor

10 Barometer

11 Air pressure sensor

12 Fingerprint sensor

13 Blood oxygen sensor

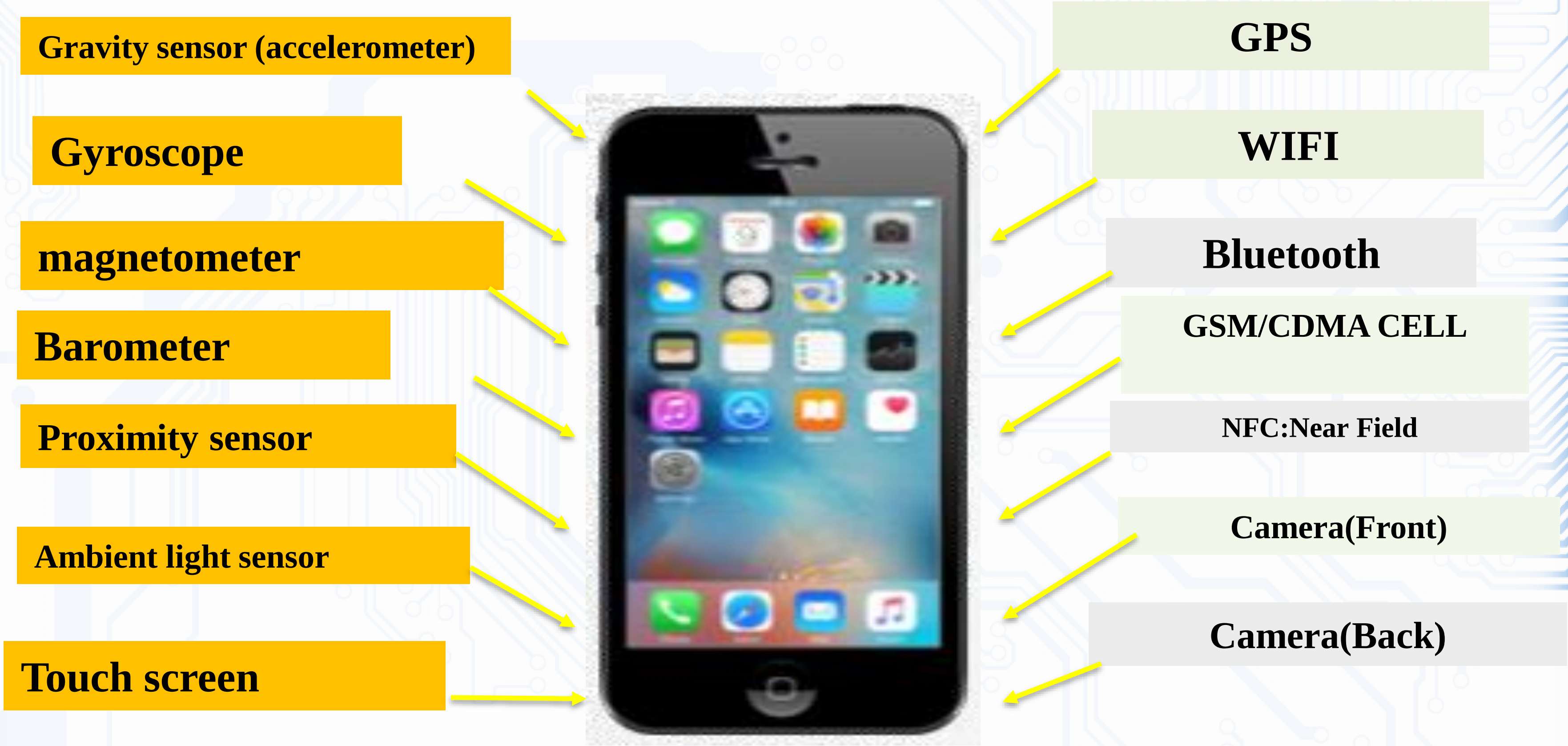
14 Heart rate sensor

15 Ultraviolet sensor

16 Posture sensor

17 Temperature Sensor

Common mobile device sensors



Gravity sensor (accelerometer)

Gyroscope

magnetometer

Barometer

Proximity sensor

Ambient light sensor

Touch screen

GPS

WIFI

Bluetooth

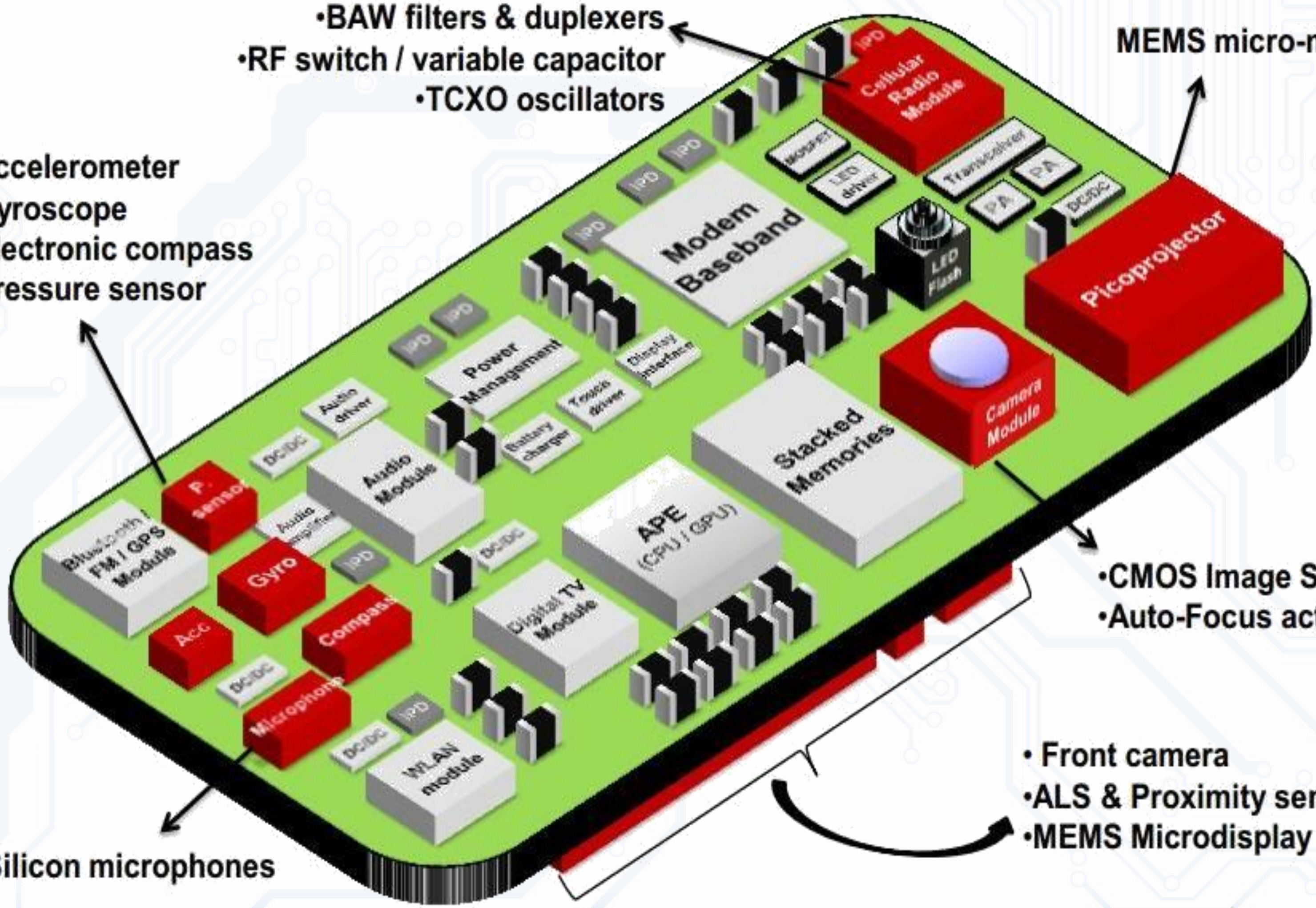
GSM/CDMA CELL

NFC:Near Field

Camera(Front)

Camera(Back)

Common mobile device sensors



- BAW filters & duplexers
- RF switch / variable capacitor
- TCXO oscillators

MEMS micro-mirror

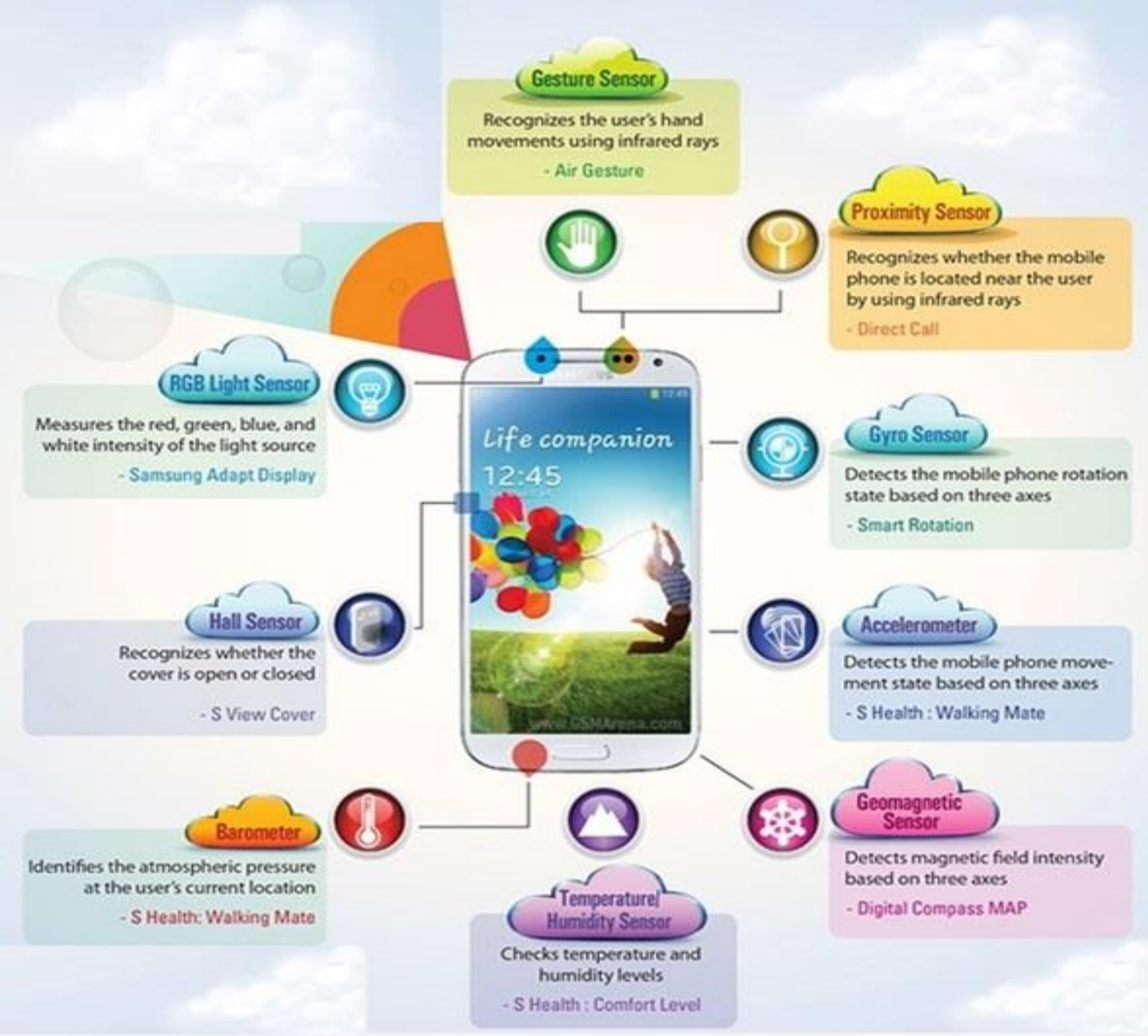
- Accelerometer
- Gyroscope
- Electronic compass
- Pressure sensor

- CMOS Image Sensor
- Auto-Focus actuator

- Front camera
- ALS & Proximity sensor
- MEMS Microdisplay

Silicon microphones

Common mobile device sensor



Gesture Sensor

Recognizes the user's hand movements using infrared rays

- Air Gesture

Proximity Sensor

Recognizes whether the mobile phone is located near the user by using infrared rays

- Direct Call

RGB Light Sensor

Measures the red, green, blue, and white intensity of the light source

- Samsung Adapt Display

Gyro Sensor

Detects the mobile phone rotation state based on three axes

- Smart Rotation

Hall Sensor

Recognizes whether the cover is open or closed

- S View Cover

Accelerometer

Detects the mobile phone movement state based on three axes

- S Health : Walking Mate

Barometer

Identifies the atmospheric pressure at the user's current location

- S Health: Walking Mate

Temperature/ Humidity Sensor

Checks temperature and humidity levels

- S Health : Comfort Level

Geomagnetic Sensor

Detects magnetic field intensity based on three axes

- Digital Compass MAP

App Inventor

App Inventor

Reading and Responding to Sensors

In this lecture , we will revisit the App Inventor components LocationSensor, OrientationSensor, and AccelerometerSensor,. Along the way, and learn about the global positioning system (GPS); **orientation measures like pitch, roll, and yaw; and some math for processing accelerometer readings.**

- Point your phone at the sky, and Google Sky Map tells you which stars you're looking at.
- Tilt your phone, and you can control the game you're playing.
- Take your phone on your daily walk, and a breadcrumb app records your route.
- **All of these apps are possible because the mobile devices we carry have high-tech sensors for detecting our location, orientation, and acceleration**

Sensors on a phone/tablet and their functions

What does a sensor do?

- A sensor is a component used in mobile devices, the purpose of which is to detect changes in the environment (such as changes in brightness, magnetic fields, temperature, and gravity) and movement (such as the device being moved, flipped, or picked up), and convert them into electronic signals that can be processed by the device.

- A mobile device sensor consists of precision components, which are sensitive to sources of external interference and physical factors.

- Avoid dropping your device or using it in environments with strong magnetic fields, abnormal humidity, extreme temperatures, or other unfavourable factors to avoid damaging its sensors.

1. Ambient light sensor

Function:

- Automatically adjusts the screen brightness of your device according to the amount of ambient light, making it more comfortable to look at the screen.
- . For example, in a particularly bright outdoor, the screen will automatically adjust to the brightest state, and in the dark environment, the screen brightness will be reduced accordingly.
- The sensor not only supports automatic white balance (AWB) when taking photos, but also works with the proximity sensor to prevent misoperations when, for example, the device is in your pocket.

How it works:

- The sensor generates strong or weak currents according to the amount of ambient light it senses, and the device increases or decreases the screen brightness accordingly.
- ***Please note that using a non-official protective case or unmatching protective film may block the ambient light sensor and affect its functions.***

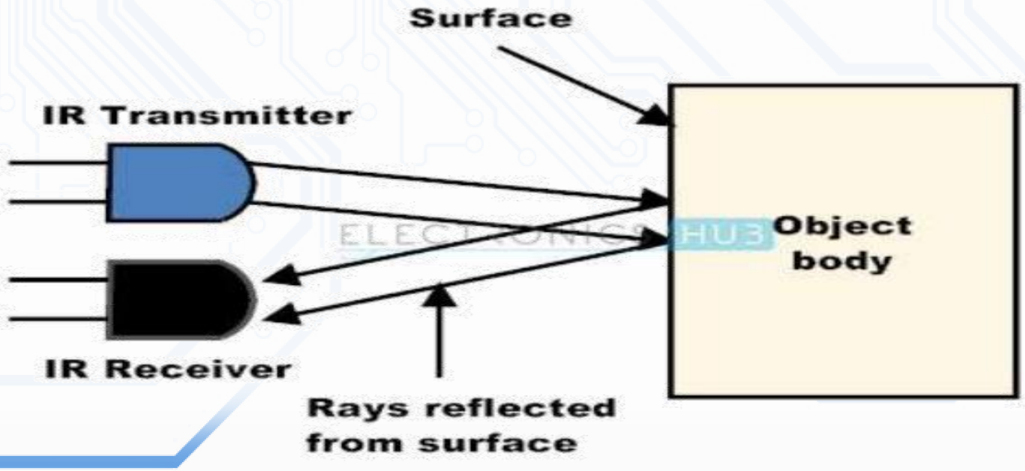
2. Proximity sensor

Function:

- Detects the presence of nearby objects.
- Equipped with a proximity sensor, your device automatically turns off the screen when it detects that it is close to your ear.
- This helps prevent possible misoperations.

How it works:

- The sensor consists of an infrared LED light and an infrared radiation (IR) detector, and is generally located at the top of the screen and near the receiver.
- It detects the distance between an object and the device by calculating changes in the infrared light signals it receives.
- The working range of a proximity sensor is generally 10 cm.



3.Distance sensor

Function:

The distance sensor is typically used in conjunction with a light sensor.

How it works:

- When you put your phone in the handset position, the distance sensor measures the distance between the phone and your ear.
- This different measurement triggers the corresponding function, such as extinguishing the screen or automatically locking the screen, and can also be used with various protective sleeves.

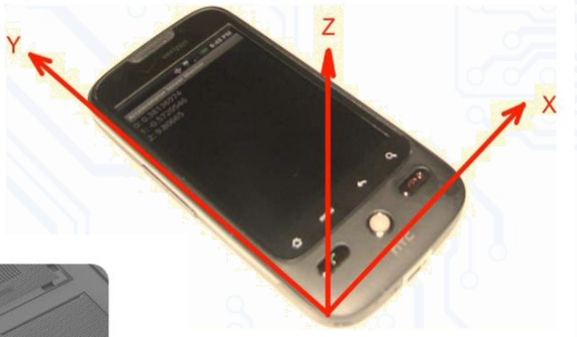
4. Gravity sensor (accelerometer)

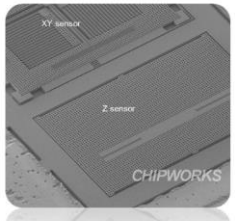
Function:

- Allows your device to automatically switch between landscape and portrait modes, count your daily steps, identify viewing orientation, use compass apps, and recognize motion gestures (such as picking up and flipping your device).
- For example, you can use your device's gravity sensor to control a moving ball or steer a race car in games, shake to switch songs, flip to mute ringtones, and so on.

How it works:

The sensor measures changes in the distance between the capacitance plates caused by motion on three axes (X, Y, and Z), and determines the instantaneous acceleration and deceleration forces accordingly.





STMicroelectronics STM331DLH three-axis accelerometer (iPhone4)

5.Acceleration sensor

Function:

- The concept of an accelerometer is slightly overlapping with the gravity sensor, but in fact it is different.
- This sensor is also used in some of the daily applications of cutting songs, flipping mute, and so on.
- .

How it works:

- The acceleration sensor is measured in multiple dimensions and mainly measures some instantaneous acceleration or deceleration.
- For example, measuring the speed of the mobile phone in the game can trigger special commands through the acceleration sensor

6. Gyroscope

Function:

- Allows you to play somatic games with your device, move your device to switch the view in games, and navigate when GPS services are not available.
- The gyroscope is also used in VR activities, 3D photography, panoramic navigation, and so on. (Some models do not have this sensor.)

How it works:

- A gyroscope is a device consisting of a rapidly spinning wheel set in a framework that permits it to tilt freely in any direction.
- The momentum of such a wheel causes it to retain its attitude when the framework is tilted, and is therefore used to measure or maintain orientation and angular velocity.
- Generally, a three-frame gyroscope is used on a device, which can simultaneously determine positions, motion tracks, and accelerations in six directions.

7. GPS position sensor

Function:

- The main function of the GPS module is to receive satellite coordinate information through the antenna to help the user locate.
- With the popularity of 4G networks, GPS is being used in more scenarios, such as remote positioning monitoring with intelligent hardware, or positioning after device loss.

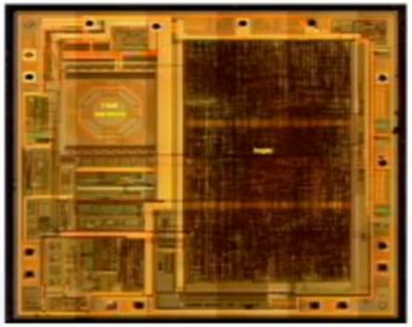
How it works:

- The mobile phone is generally equipped with A-GPS.
- The so-called A-GPS is a faster positioning through the mobile network based on receiving the navigation satellite signal, which is more advanced than the ordinary GPS.

8. Compass

Function:

- Provides compass and map navigation functions to help locate positions more accurately. (Some models do not have this sensor.)



3-Axis Compass

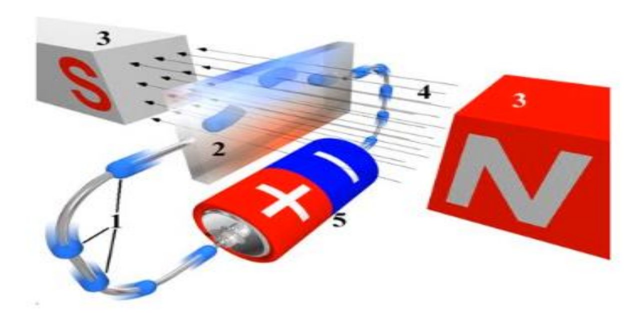
How it works:

- The Hall or magnetoresistance principle detects the size and direction of the magnetic field, and determines the magnetic field strength based on the three-axis reading of the sensor.
- The direction of the device can then be determined. Sometimes you need to shake or rotate your device for the compass to work properly.
- To ensure the accuracy of measurement results, avoid placing your device with magnetic objects.

9. Hall effect sensor

Function:

- Used in Smart cover mode, where your device automatically locks the screen when you close the flip cover, and unlocks the screen when you open the flip cover. (Some models do not have this sensor.)



Hall Effect

How it works:

- In a Hall effect sensor, a conductor has a current applied along it. When the conductor is placed in a magnetic field perpendicular to the direction of the electrons, they will be deflected from a straight path.
- As a consequence, one plane of the conductor will become negatively charged and the opposite side will become positively charged, resulting in voltage differences.
- Please note that the sensor functions may be affected if you use a non-official flip cover.

10. Barometer

Function:

- Corrects altitude measurement errors to narrow down the deviation to about 1 meter and works with your device's GPS to find your altitude when you are under an overpass or inside a high-rise building. (Some models do not have this sensor.)

How it works:

- Consisting of a rheostat and a capacitor, a barometer measures the atmospheric pressure by calculating changes in electrical resistance and capacitance.

11. Air pressure sensor

Function:

- The air pressure sensor has been used in military mobile phones before, and is divided into a variable displacement air pressure sensor and a variable resistance air pressure sensor.
- A change in air pressure causes a change in the resistance or capacitance measurement.
- General GPS can calculate your position, but for some height changes you need a barometric pressure sensor to measure.

How it works:

- A mobile phone with this sensor can measure how many floors you have on a day, or for indoor positioning, and the internal air pressure sensor is mainly the degree of test equipment closure.

12. Fingerprint sensor

Function:

- Since 2013, fingerprint sensors have exploded in smartphones. It can automatically collect users’ fingerprints for privacy protection purposes.
- However, mobile phones with fingerprint sensors are not only unlocking devices, but also interacting with mobile payments, including Apple Pay and Sumsang Pay, all based on fingerprint sensors.
- Verifies your fingerprint for screen unlock and making payments.
-

How it works:

- Capacitive fingerprint sensors, commonly used on devices, sense the electrical current produced by finger touches and generate an image of the ridges and valleys that make up a fingerprint. The sensor then compares the image with the stored version on the device.
- In-screen fingerprint sensors (optical fingerprint sensors) sense the fingerprints using light reflections. They are used with OLED screens where spacing between pixels of an OLED allows for light transmission. When a user touches the fingerprint sensor icon, the OLED illuminates the touched area. The in-screen sensor underneath the screen then grabs an image of the fingerprint that is projected onto the sensor. The sensor then compares the image with the stored version on the device.
-

13. Ultraviolet sensor

Function:

- The ultraviolet sensor uses the photoelectric emission effect to measure, and the outdoor light source is taken by the camera to convert the ultraviolet light intensity into a discharge effect.
- Mobile phones using such sensors are rare, and the stability of the measurements is subject to further observation.

14. Heart rate sensor

Function:

- The heart rate sensor is more common in the wearable device, but the application on the mobile phone is generally set at the back of the mobile phone.

How it works:

- The high-intensity LED light source illuminates the finger to convert the corresponding data to the heart rate, and the finger needs to be kept stable during the test. Otherwise, the test results will have a large deviation.

15. Blood oxygen sensor

Function:

- Like the heart rate sensor, the hemoglobin and oxyhemoglobin in the blood absorb different amounts of infrared light and red light.

How it works:

- When the two LED lights of the infrared light and the red light are simultaneously irradiated with the finger, the absorption of the reflected light can also be measured.
- The spectrum is thus measured for blood oxygen content.

16. Posture sensor

Function:

- The posture sensor is widely used in the unmanned aerial vehicle (droned), robot, mechanical pan-tilt-zoom (PTZ), vehicle and ship, ground and underwater devices, and human body motion analysis devices that need toW measure the three-dimensional posture and orientation.

How it works:

- A posture sensor is a high-performance 3D motion posture measurement system based on the micro-electro-mechanical systems (MEMS) technology.
- It includes a triaxial gyroscope, triaxial accelerometer, triaxial electronic compass, or other motion sensors, and obtains data such as a three-dimensional posture and orientation after temperature compensation by using an embedded low-power
- ARM processor. Based on quaternion algorithm and special data fusion technology, the zero-drift three-dimensional posture orientation data represented by quaternion and Euler angle are output in real time.

17.Temperature Sensor

Function:

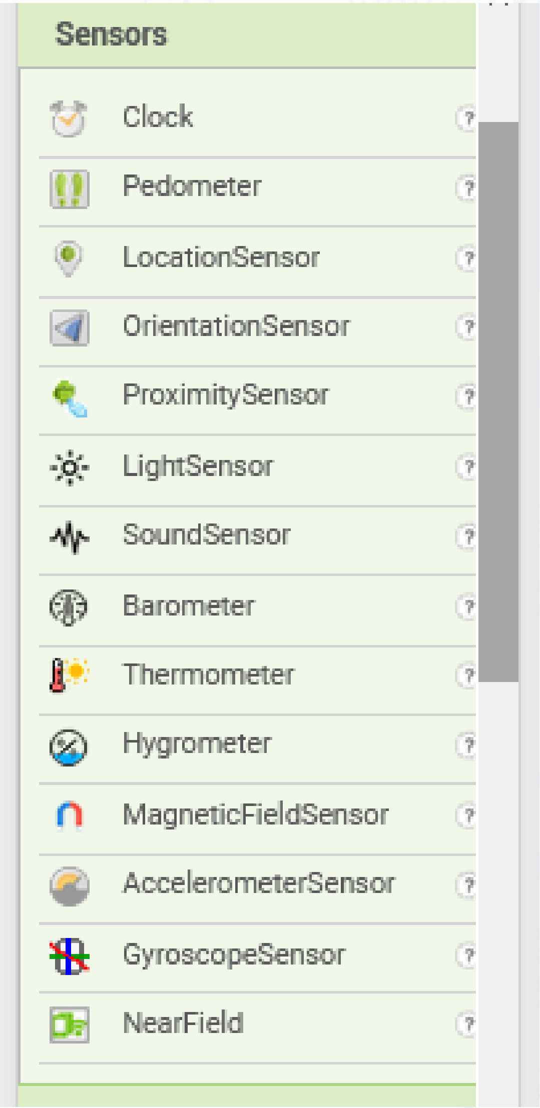
- The temperature sensor is used to detect the temperature change of the mobile phone itself, and the degree of heat generation of the mobile phone can be seen.

How it works:

- In terms of extended functions, the temperature sensor can also detect temperature changes in the outside air, even the user's current body temperature.

Sensors

- Accelerometer Sensor
- Barcode Scanner
- Barometer
- Clock
- Gyroscope Sensor
- Hygrometer
- Light Sensor
- Location Sensor
- MagneticField Sensor
- NearField
- Orientation Sensor
- Pedometer
- Proximity Sensor
- Thermometer



App Inventor

Accelerometer Sensor

- Non-visible component that can detect shaking and measure acceleration approximately in three dimensions using SI units (m/s2).
- The components are:
  - xAccel:
    - **0 when the phone is at rest on a flat surface, positive when the phone is tilted to the right (i.e., its left side is raised),**
    - **and negative when the phone is tilted to the left (i.e., its right size is raised).**
  - yAccel:
    - **0 when the phone is at rest on a flat surface,**
    - **positive when its bottom is raised, and negative when its top is raised.**
  - zAccel:
    - Equal to -9.8 (earth’s gravity in meters per second per second when the device is at rest parallel to the ground with the display facing up,
    - 0 when perpendicular to the ground, and +9.8 when facing down.
    - The value can also be affected by accelerating it with or against gravity.

Barcode Scanner

Component for scanning a QR code and getting back the resulting string.

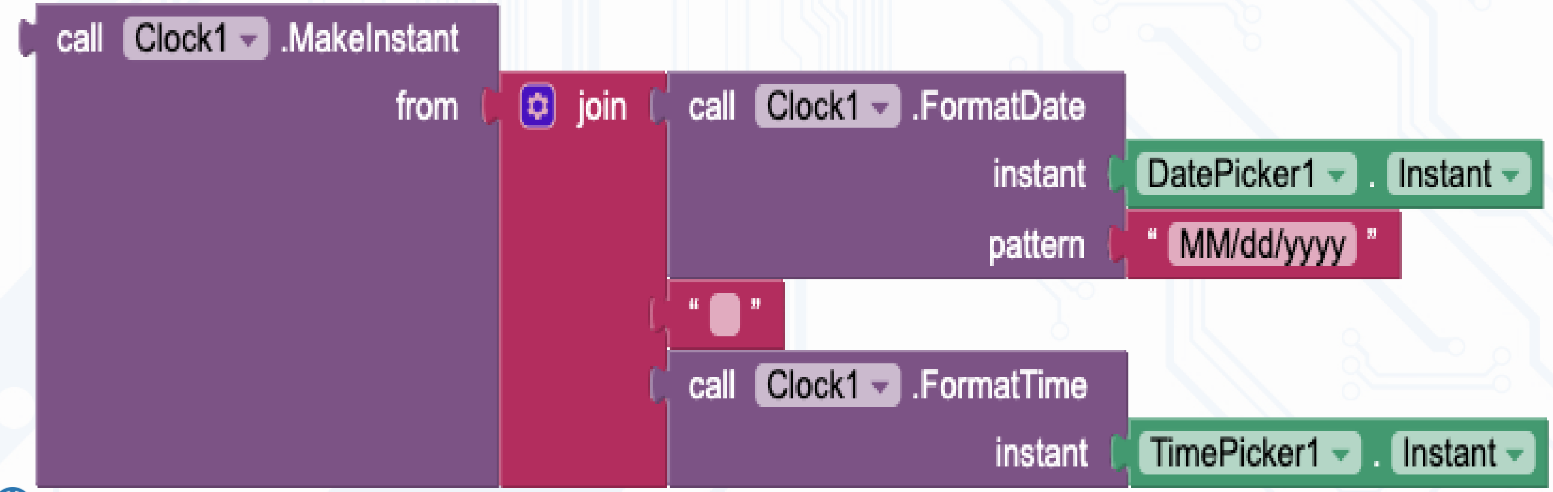
Properties	
Result	Gets the text result of the previous scan.
UseExternalScanner:	Set whether or not you wish to use an External Scanning program such as Bar Code Scanner. If false a version of ZXing integrated into App Inventor will be used.

Events	
AfterScan(result)	Indicates that the scanner has read a (text) result and provides the result

Methods	
DoScan()	Begins a barcode scan, using the camera. When the scan is complete, the AfterScan event will be raised.

Clock

- Non-visible component that provides the instant in time using the internal clock on the phone. It can fire a timer at regularly set intervals and perform time calculations, manipulations, and conversions.
- Operations on dates and times, such as from DatePicker and TimePicker, are accomplished through methods in Clock. Date and Time are represented as InstantInTime and Duration.
  - Instant: consists of Year, Month, DayOfMnoth, Hour, Minute, and SEcond. An instant can be created using the MakeInstant, MakeInstantFromMillis and MakeInstantFromParts methods.
  - Duration: time in milliseconds elapsed between instants. Duration can be obtained by the

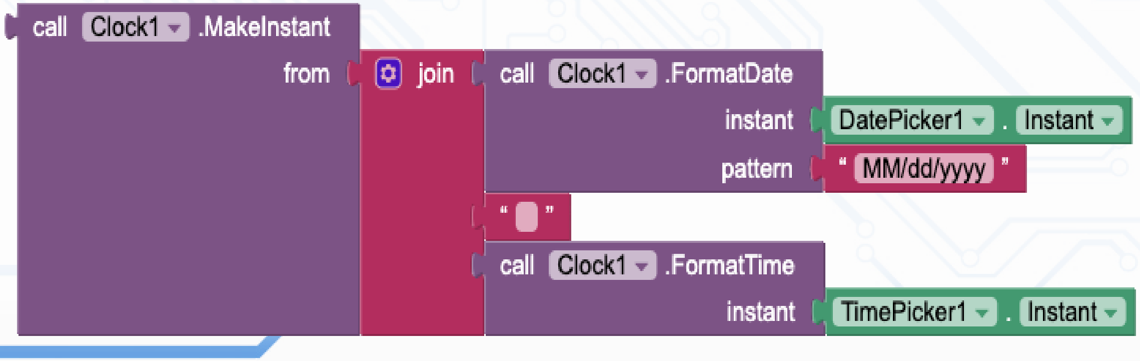


```
call Clock1 .MakeInstant
    from  join  call Clock1 .FormatDate
                    instant  DatePicker1 . Instant
                    pattern  " MM/dd/yyyy "
                " "
                call Clock1 .FormatTime
                    instant  TimePicker1 . Instant
```

Clock

- Duration method.
  - Instants are assumed to be in the device’s local time zone. When they are converted to or from milliseconds, the milliseconds for a given Instance are calculated from January 1, 1970 in UTC (Greenwich Mean Time).
  - Methods to convert an Instant to text are also available.
  - Acceptable patterns are empty string, MM/dd/YYYY HH:mm:ss a, or MMM d, yyyy HH:mm.
  - The empty string will provide the default format, which is "MMM d, yyyy HH:mm:ss a" for FormatDateTime, "MMM d, yyyy" for FormatDate. To see all possible formats, please see here.
  - A note on combining date and time:

In order to combine the date from one Instant and the time from another, for example from a DatePicker and TimePicker, extract the parts as text and use the text to create a new Instant.



```
call Clock1 .MakeInstant
    from  join  call Clock1 .FormatDate
                    instant  DatePicker1 . Instant
                    pattern  " MM/dd/yyyy "
                " "
                call Clock1 .FormatTime
                    instant  TimePicker1 . Instant
```

Gyroscope Sensor

- Component providing data from the device’s gyroscope sensor.

Properties	
Available	Indicates whether a gyroscope sensor is available.
Enabled	Enabled property getter method.
XAngularVelocity	The angular velocity around the X axis, in degrees per second.
YAngularVelocity	The angular velocity around the Y axis, in degrees per second.
ZAngularVelocity	The angular velocity around the Z axis, in degrees per second.

Events
GyroscopeChanged(xAngularVelocity,yAngularVelocity,zAngularVelocity,timestamp)
Indicates that the gyroscope sensor data has changed. The timestamp parameter is the time in nanoseconds at which the event occurred.

Hygrometer

- Physical world component that can measure the relative ambient air humidity if supported by the hardware.

Light Sensor

- Physical world component that can measure the light level.

Properties	
Available	Specifies whether or not the device has the hardware to support the LightSensor component.
AverageLux:	Returns the brightness in lux by averaging the previous 10 measured values. The sensor must be enabled and available to return meaningful values.
Enabled	Specifies whether the sensor should generate events. If true, the sensor will generate events. Otherwise, no events are generated.
Lux	Returns the last measured brightness in lux. The sensor must be enabled and available to return meaningful values.
RefreshTime	The requested minimum time in milliseconds between changes in readings being reported. Android is not guaranteed to honor the request. Setting this property has no effect on pre-Gingerbread devices.

Events	
LightChanged(*lux*):	Indicates the light level changed.

LocationSensor

- Non-visible component providing location information, including Latitude, Longitude, Altitude (if supported by the device), speed (if supported by the device), and address.
- **This can also perform “geocoding”, converting a given address (not necessarily the current one) to a latitude (with the LatitudeFromAddress method) and a longitude (with the LongitudeFromAddress method).**
- In order to function, the component must have its Enabled property set to true, and the device must have location sensing enabled through wireless networks or GPS satellites (if outdoors).
- Location information might not be immediately available when an app starts.
- You’ll have to wait a short time for a location provider to be found and used, or wait for the LocationChanged event.
- **The emulator does not emulate sensors on all devices. Code should be tested on a physical device.**

MagneticField Sensor

- Component for MagneticFieldSensor

Properties	
AbsoluteStrength	Indicates the absolute strength of the field.
Enabled	Indicates whether or not the magnetic field sensor is enabled and working.
MaximumRange	Indicates the maximum range the magnetic sensor can reach.
XStrength	Indicates the field’s strength in the X-axis.
YStrength	Indicates the field’s strength in the Y-axis.
ZStrength	Indicates the field’s strength in the Z-axis.

Events	
MagneticChanged(xStrength,yStrength,zStrength,absoluteStrength)	Triggers when magnetic field has changed, setting the new values in parameters.

NearField

- Non-visible component to provide NFC capabilities. For now this component supports the reading and writing of text tags only (if supported by the device).
- In order to read and write text tags, the component must have its ReadMode property set to true or false respectively.
- **Note: This component will only work on Screen1 of any App Inventor app.**

Orientation Sensor

- **Use an orientation sensor component to determine the phone’s spatial orientation.** An orientation sensor is a non-visible component that reports the following three values, in degrees:
  - Roll :
    - 0 degree when the device is level,
    - increasing to 90 degrees as the device is tilted up onto its left side, and decreasing to −90 degrees when the device is tilted up onto its right side.
  - Pitch :
    - 0 degree when the device is level, increasing to 90 degrees as the device is tilted so its top is pointing down, then decreasing to 0 degree as it gets turned over.
    - Similarly, as the device is tilted so its bottom points down, pitch decreases to −90 degrees, then increases to 0 degree as it gets turned all the way over.
  - Azimuth :
    - 0 degree when the top of the device is pointing north,
    - 90 degrees when it is pointing east,
    - 180 degrees when it is pointing south,
    - 270 degrees when it is pointing west, etc.
- These measurements assume that the device itself is not moving.

Pedometer

- This component keeps count of steps using the accelerometer.

Properties	
Distance	Returns the approximate distance traveled in meters.
ElapsedTime	Returns the time elapsed in milliseconds since the pedometer has started.
SimpleSteps	Returns the number of simple steps taken since the pedometer has started.
StopDetection Timeout	Returns the duration of idleness (no steps detected) after which to go into a “stopped” state.
StrideLength	Returns the current estimate of stride length in meters, if calibrated, or returns the default (0.73 m) otherwise.
WalkSteps	Returns the number of walk steps taken since the pedometer has started.

Events	
SimpleStep(simpleSteps,distance)	This event is run when a raw step is detected.
WalkStep(walkSteps,distance)	This event is run when a walking step is detected. A walking step is a step that appears to be involved in forward motion.

Proximity Sensor

- A sensor component that can measure the proximity of an object (in cm) relative to the view screen of a device.
- This sensor is typically used to determine whether a handset is being held up to a persons ear; i.e. lets you determine how far away an object is from a device. Many devices return the absolute distance, in cm, but some return only near and far values.
- In this case, the sensor usually reports its maximum range value in the far state and a lesser value in the near state.
- It reports the following value:

  Distance: The distance from the object to the device

Thermometer

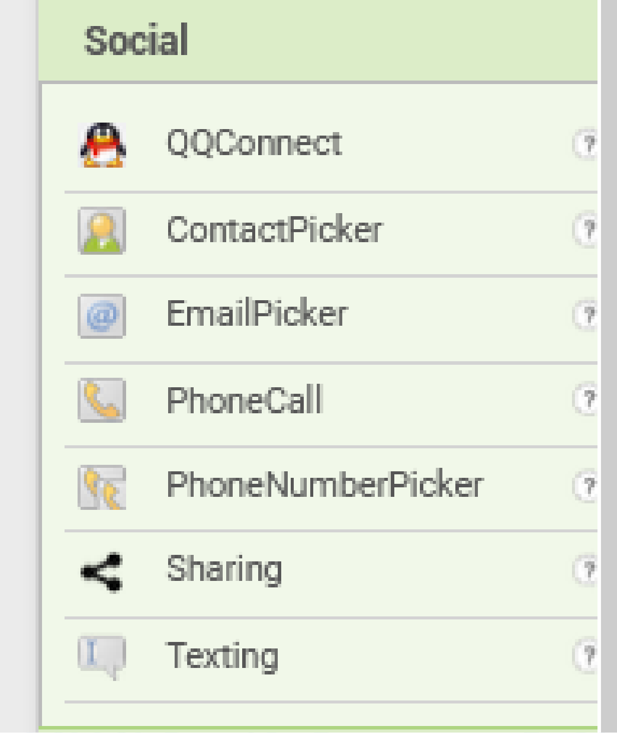
Physical world component that can measure the ambient air temperature if supported by the hardware.

Properties	
Available	Specifies whether or not the device has the hardware to support the Thermometer component.
Enabled	Specifies whether the sensor should generate events. If true, the sensor will generate events. Otherwise, no events are generated.
RefreshTime	The requested minimum time in milliseconds between changes in readings being reported. Android is not guaranteed to honor the request. Setting this property has no effect on pre-Gingerbread devices.
Temperature	Returns the temperature in degrees Celsius. The sensor must be enabled and available to return meaningful values.

Events	
TemperatureChanged(temperature)	Indicates a change of temperature, provided in degrees Celsius.

Social

- Table of Contents:
  - ContactPicker
  - EmailPicker
  - PhoneCall
  - PhoneNumberPicker
  - Sharing
  - Texting
  - Twitter



Social

- QQConnect
- ContactPicker
- EmailPicker
- PhoneCall
- PhoneNumberPicker
- Sharing
- Texting

Contact Picker

- A button that, when clicked on, displays a list of the contacts to choose among.
- After the user has made a selection, the following properties will be set to information about the chosen contact:
  - ContactName: the contact’s name
  - EmailAddress: the contact’s primary email address
  - EmailAddressList: a list of the contact’s email addresses
  - ContactUri: the contact’s URI on the device
  - PhoneNumber: the contact’s primary phone number (on Later Android Verisons)
  - PhoneNumberList: a list of the contact’s phone numbers (on Later Android Versions)
  - Picture: the name of the file containing the contact’s image, which can be used as a Picture property value for the Image or ImageSprite component.
- Other properties affect the appearance of the button (TextAlignment, BackgroundColor, etc.) and whether it can be clicked on (Enabled).
- **The ContactPicker component might not work on all phones.**
- **For example, on Android systems before system 3.0, it cannot pick phone numbers, and the list of email addresses will contain only one email.**

EmailPicker

- An EmailPicker is a kind of text box. If the user begins entering the name or email address of a contact, the phone will show a dropdown menu of choices that complete the entry.
- If there are many contacts, the dropdown can take several seconds to appear, and can show intermediate results while the matches are being computed.
- The initial contents of the text box and the contents< after user entry is in the Text property.
- If the Text property is initially empty, the contents of the Hint property will be faintly shown in the text box as a hint to the user.
- Other properties affect the appearance of the text box (TextAlignment, BackgroundColor, etc.) and whether it can be used (Enabled).
- **Text boxes like this are usually used with Button components, with the user clicking on the button when text entry is complete.**

PhoneCall

- **A non-visible component that makes a phone call to the number specified in the PhoneNumber property, which can be set either in the Designer or Blocks Editor.**
- The component has a MakePhoneCall method, enabling the program to launch a phone call. You may also use MakePhoneCallDirect to directly initiate a phone call without user interaction.
- However, apps using this block may require further review by Google if submitted to the Play Store so it is advised to use MakePhoneCall instead.
- Often, this component is used with the ContactPicker component, which lets the user select a contact from the ones stored on the phone and sets the PhoneNumber property to ContactPicker’s PhoneNumber property.
- **To directly specify the phone number (e.g., 650-555-1212), set the PhoneNumber property to a Text with the specified digits (e.g., “6505551212”). Dashes, dots, and parentheses may be included (e.g., “(650)-555-1212”) but will be ignored; spaces may not be included.**

PhoneNumberPicker

- A button that, when clicked on, displays a list of the contacts' phone numbers to choose among.
- After the user has made a selection, the following properties will be set to information about the chosen contact:
  - ContactName: the contact's name
  - PhoneNumber: the contact's phone number
  - EmailAddress: the contact's email address
  - Picture: the name of the file containing the contact's image, which can be used as a Picture property value for the Image or ImageSprite component.
- Other properties affect the appearance of the button (TextAlignment, BackgroundColor, etc.) and whether it can be clicked on (Enabled).
- The PhoneNumberPicker component may not work on all Android devices. For example, on Android systems before system 3.0, the returned lists of phone numbers and email addresses will be empty.

Sharing

- Sharing is a non-visible component that enables sharing files and/or messages between your app and other apps installed on a device.
- The component will display a list of the installed apps that can handle the information provided, and will allow the user to choose one to share the content with, for instance a mail app, a social network app, a texting app, and so on.
- The file path can be taken directly from other components such as the Camera or the ImagePicker, but can also be specified directly to read from storage.
- **Be aware that different devices treat storage differently, so a few things to try if, for instance, you have a file called arrow.gif in the folder Appinventor/assets, would be:**
- **"file:///sdcard/Appinventor/assets/arrow.gif";**
- **or"/storage/Appinventor/assets/arrow.gif"**

App Inventor

Texting

- A component that will, when the SendMessage method is called, launch the device’s preferred texting app to send the text message specified in the SendMessage property to the phone number specified in the PhoneNumber property.
- You may also send text messages without user interaction by calling SendMessageDirect instead, but this adds dangerous permissions to your final app.
  - If the ReceivingEnabled property is set to 1 messages will not be received.
  - If ReceivingEnabled is set to 2 messages will be received only when the application is running.
  - Finally if ReceivingEnabled is set to 3, messages will be received when the application is running and when the application is not running they will be queued and a notification displayed to the user.
- When a message arrives, the MessageReceived event is raised and provides the sending number and message.

Texting

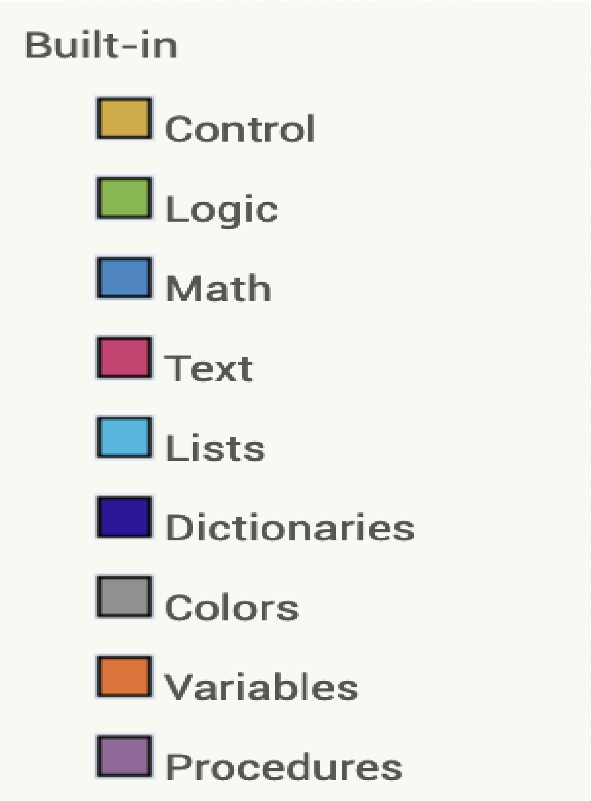
- An app that includes this component will receive messages even when it is in the background (i.e. when it’s not visible on the screen) and, moreso, even if the app is not running, so long as it’s installed on the phone.
- If the phone receives a text message when the app is not in the foreground, the phone will show a notification in the notification bar. Selecting the notification will bring up the app.
- As an app developer, you’ll probably want to give your users the ability to control ReceivingEnabled so that they can make the phone ignore text messages.
- If the GoogleVoiceEnabled property is true, messages can be sent over Wifi using Google Voice. This option requires that the user have a Google Voice account and that the mobile Voice app is installed on the phone.
- The Google Voice option works only on phones that support Android 2.0 (Eclair) or higher. Unfortunately, receiving no longer works in Google Voice due to changes introduced in Google Voice App.
- To specify the phone number (e.g., 650-555-1212), set the PhoneNumber property to a Text string with the specified digits (e.g., 6505551212). Dashes, dots, and parentheses may be included (e.g., (650)-555-1212) but will be ignored; spaces may not be included.
- Another way for an app to specify a phone number would be to include a PhoneNumberPicker component, which lets the users select a phone numbers from the ones stored in the the phone’s contacts.

Twitter

- **A non-visible component that enables communication with Twitter. Once a user has logged into their Twitter account (and the authorization has been confirmed successful by the IsAuthorized event), many more operations are available:**
  - Searching Twitter for tweets or labels (SearchTwitter)
  - Sending a Tweet (Tweet)
  - Sending a Tweet with an Image (TweetWithImage)
  - Directing a message to a specific user (DirectMessage)
  - Receiving the most recent messages directed to the logged-in user (RequestDirectMessages)
  - Following a specific user (Follow)
  - Ceasing to follow a specific user (StopFollowing)
  - Getting a list of users following the logged-in user (RequestFollowers)
  - Getting the most recent messages of users followed by the logged-in user (RequestFriendTimeline)
  - Getting the most recent mentions of the logged-in user (RequestMentions)
- You must obtain a Consumer Key and Consumer Secret for Twitter authorization specific to your app from http://twitter.com/oauth_clients/new

Built-in blocks

- Built-in blocks are available regardless of which components are in your project.
- In addition to these language blocks, each component in your project has its own set of blocks specific to its own events, methods, and properties.
- This is an overview of all of the Built-In Blocks available in the Blocks Editor.

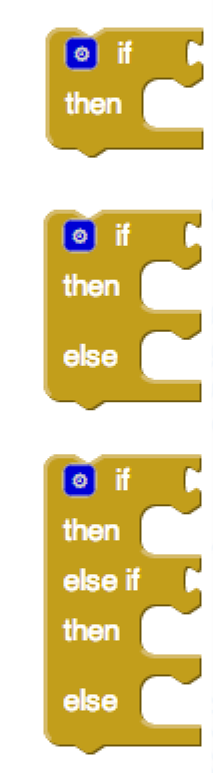


Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

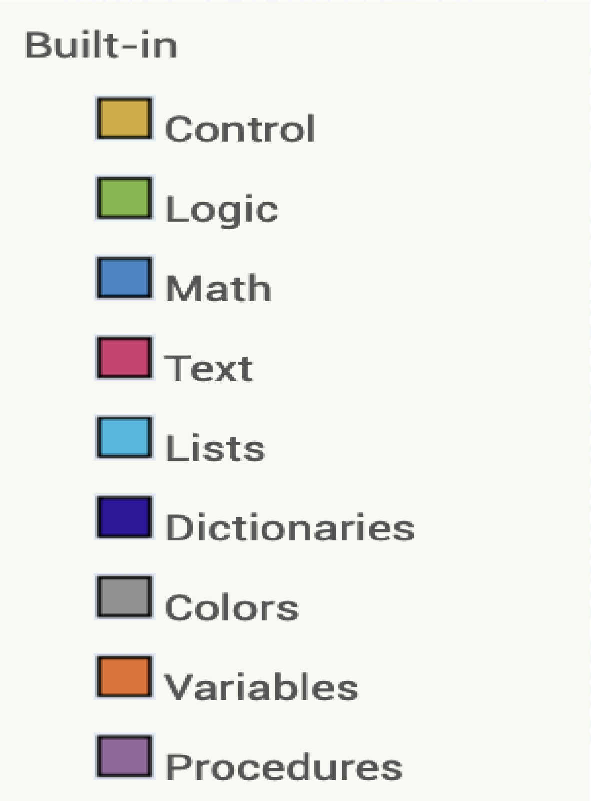
MIT App Inventor Control Blocks

- if & else if
- for each number from to
- for each item in list
- for each key with value in dictionary
- While
- if then else
- do with result
- evaluate but ignore result
- open another screen
- open another screen with start value
- get plain start text
- get start value
- close screen
- close screen with plain text
- close screen with value
- close application
- break



MIT App Inventor Logic Blocks

- True
- False
- Not
- =
- ≠
- And
- or



Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

MIT App Inventor Math Blocks

- 0 (basic number block)
- 0 (radix number block)
- =
- ≠
- >
- ≥
- <
- ≤
- +
- -
- *
- /
- [^](#exponent)
- random integer

- random integer
- random fraction
- random set seed to
- min
- max
- sqrt
- abs
- neg
- log
- e^
- round
- ceiling
- floor
- modulo
- remainder
- quotient

- sin
- cos
- tan
- asin
- acos
- atan
- atan2
- convert radians to degrees
- convert degrees to radians
- format as a decimal
- is a number
- convert number
- bitwise and
- bitwise or (inclusive)
- bitwise or (exclusive

- string
- join
- length
- is empty
- compare texts
- trim
- Upcase
- downcase
- starts at
- Contains
- contains any
- contains all

- split at first
- split at first of any
- split
- split at any
- split at spaces
- segment
- replace all
- obfuscated text
- is a string?
- reverse
- replace all mappings

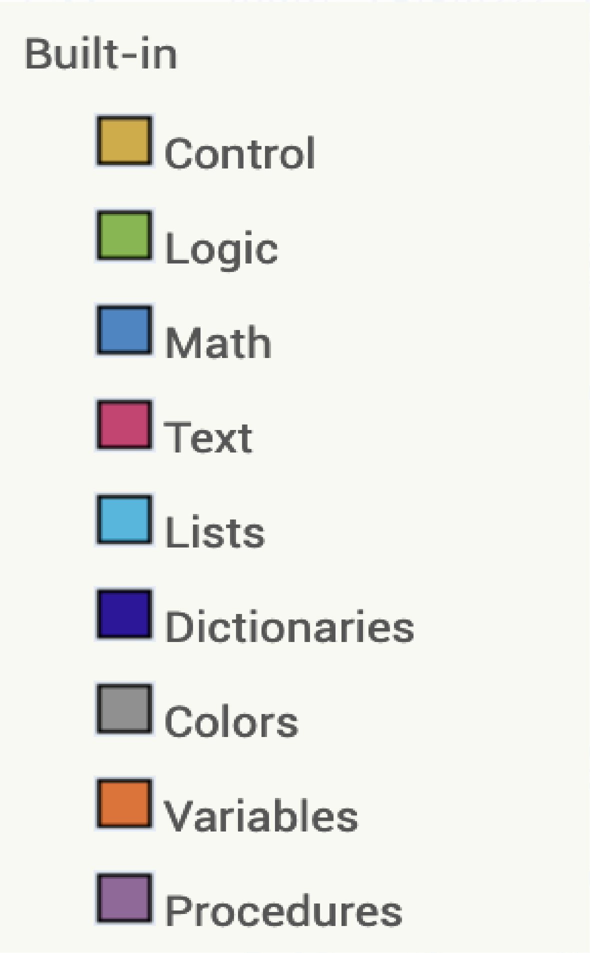
MIT App Inventor List Blocks

- create empty list
- make a list
- add items to list
- is in list
- length of list
- is list empty
- pick a random item
- index in list
- select list item
- insert list item
- replace list item
- remove list item
- append to list
- copy list
- is a list?
- reverse list
- list to csv row
- list to csv table
- list from csv row
- list from csv table
- lookup in pairs
- join with separator

- Dictionaries blocks
- Introduction
- create empty dictionary
- make a dictionary
- pair
- get value for key
- set value for key
- delete entry for key
- get value at key path
- set value for key path
- get keys
- get values
- is key in dictionary?
- size of dictionary
- list of pairs to dictionary
- dictionary to list of pairs
- copy dictionary
- merge into dictionary
- list by walking key path
- walk all at level
- is a dictionary?

Colors blocks

- a color box
- make color
- split color
- Variables blocks
- initialize global name to
- Get
- Set
- initialize local name to in (do)
- initialize local name to in (return)
- Procedures blocks
- procedure do
- procedure result



Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

@Student Task

- Check the sensor section in MIT app

- JUST ON MOOC and just PPT format
- Just format
- Your file should have this format of name

<Task number><student name><Student ID>.ppt

Reference

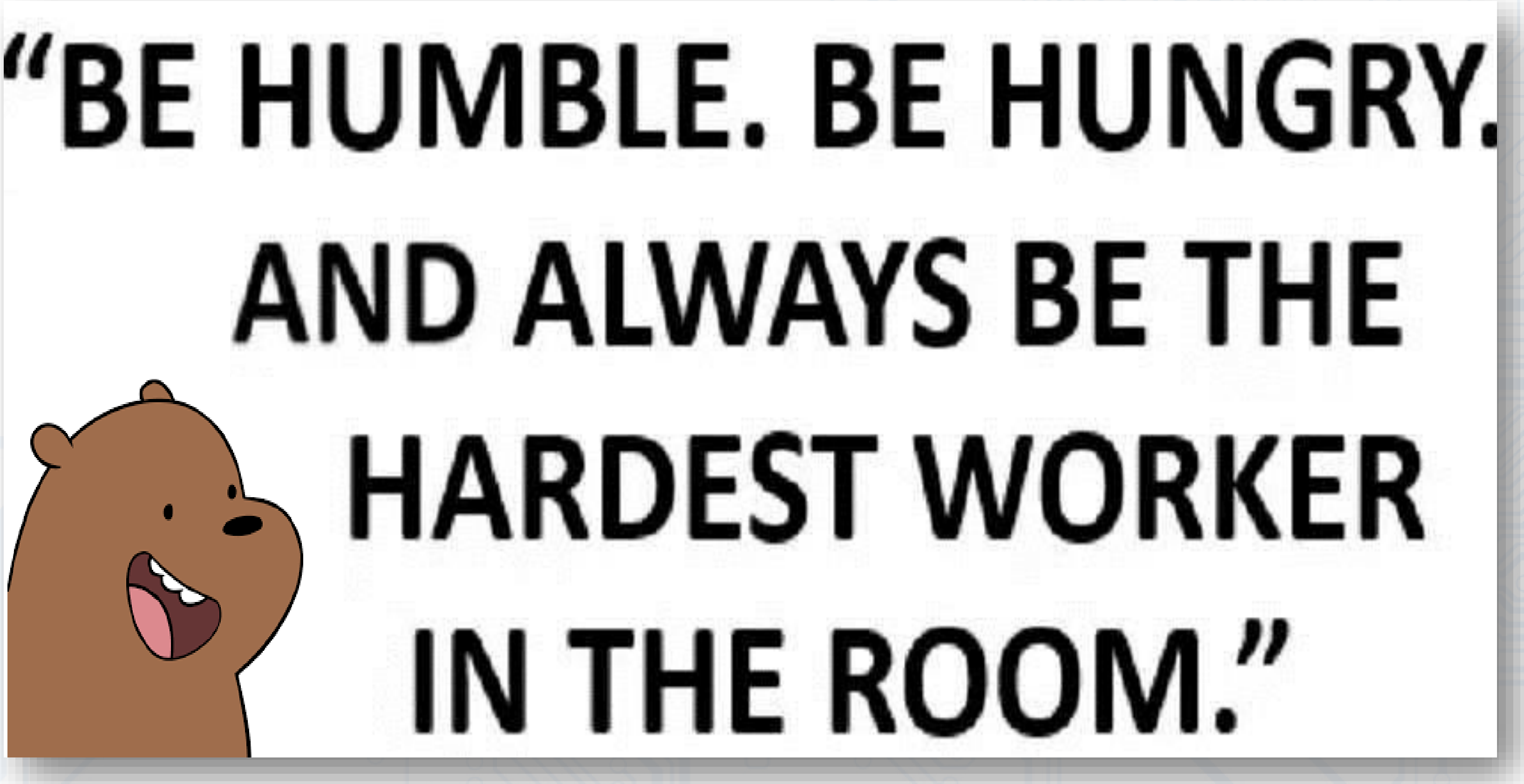
- **https://magora-systems.com/application-development-life-cycle/**
- **Teaching with AppInventor**

  http://appinventor.mit.edu/explore/teach.html
- **AppInventor Tutorials:**

  http://appinventor.mit.edu/explore/ai2/tutorials.html
- **Sounds** http://www.soundbible.com
- **App Inventor:** http://appinventor.googlelabs.com/
- **Appinventor.org:** http://www.appinventor.org/
- **Wolber, Abelson et al. text:** http://www.appinventor.org/text2011
- **Group:** http://groups.google.com/group/app-inventor-instructors
- **Wolber course:** http://appinventor.org/course-in-a-box
- **Morelli course:** http://turing.cs.trincoll.edu/~ram/cpsc110/

Reference

- **Teaching with AppInventor** http://appinventor.mit.edu/explore/teach.html **AppInventor Tutorials:** http://appinventor.mit.edu/explore/ai2/tutorials.html
- **Sounds** http://www.soundbible.com
- **App Inventor:** http://appinventor.googlelabs.com/
- **Appinventor.org:** http://www.appinventor.org/
- **Wolber, Abelson et al. text:** http://www.appinventor.org/text2011
- **Group:** http://groups.google.com/group/app-inventor-instructors
- **Wolber course:** http://appinventor.org/course-in-a-box
- **Morelli course:** http://turing.cs.trincoll.edu/~ram/cpsc110/



"BE HUMBLE. BE HUNGRY. AND ALWAYS BE THE HARDEST WORKER IN THE ROOM."

江西理工大学

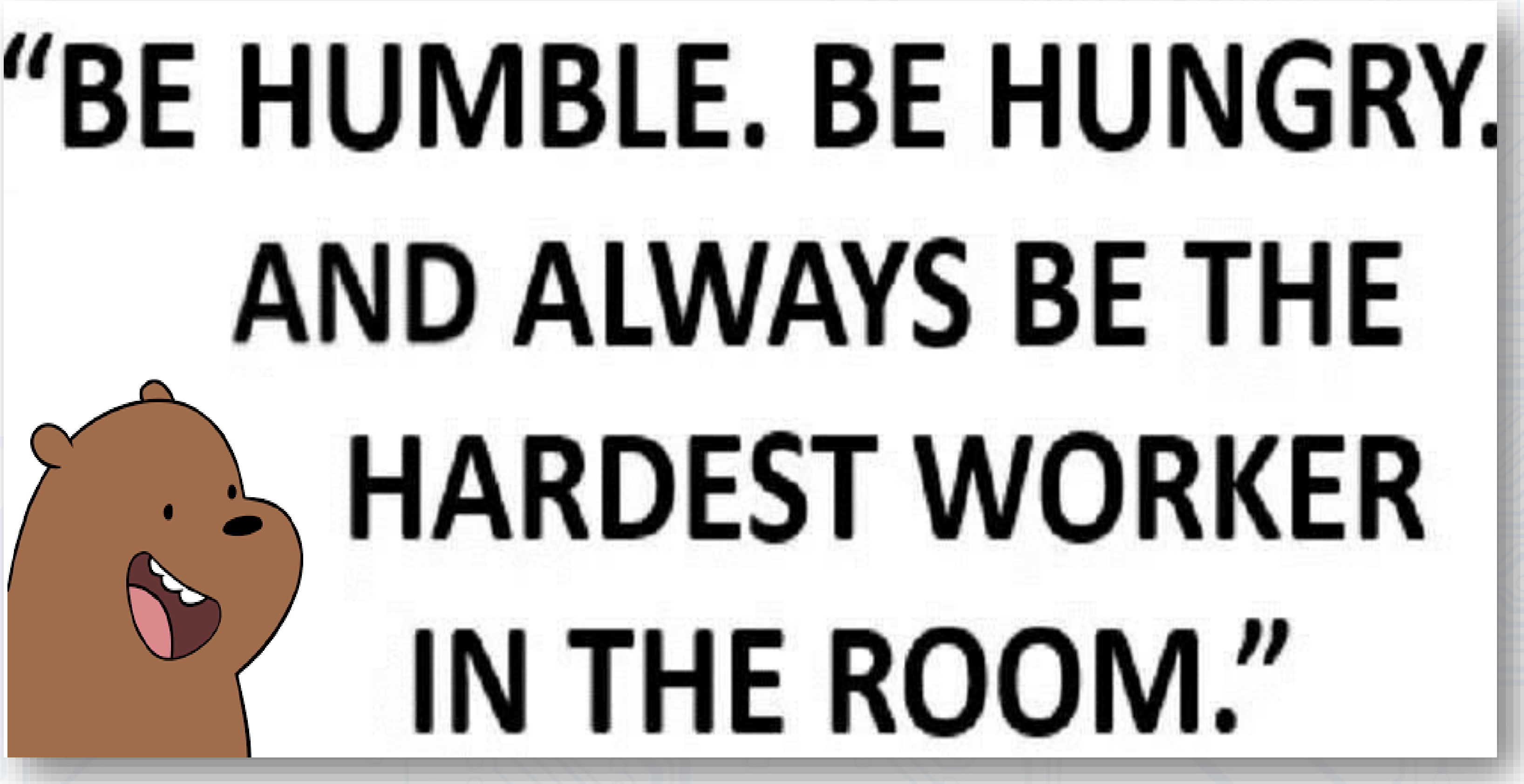
Jiangxi University of Science and Technology

信息工程学院

School of information engineering

Digital Image Processing

THANK YOU



"BE HUMBLE. BE HUNGRY. AND ALWAYS BE THE HARDEST WORKER IN THE ROOM."