🎓 AB4.2 – App Developer

Mini task Day 4

Project: The Red Square Thinks With Us
Role: App Developer – test, improve, prepare the presentation

Goal today

The app replaces the If/Else rule with an ML4K Numbers model:

- temperature (String) receive
- per ML4K als cool/normal/hot klassifizieren
- **Unsigned-Short an Arduino send: 1=hot (Red/fog an), 2=cool (Green), 3=normal (Yellow)**

Needed

- App Inventor + AI2 Companion
- Bluetooth & Standort erlaubt (Ondroid)
- **ML4K extension in your project (e.g. ML4K1) + ML4K key**
- Globale variablen: temperatureReading (number), canWrite (Bool, false), (optional) prev_cmd (number)

Sequence of steps & blocks

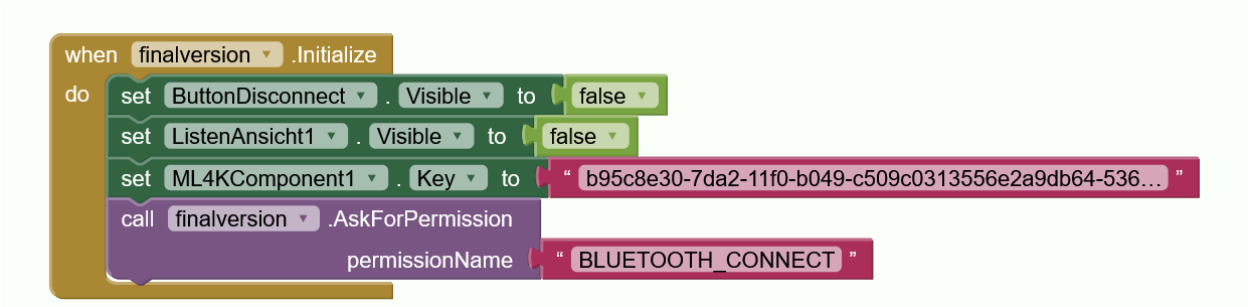
1) Connect & Schreibrecht check

- In the Connected event, set the status and check whether UUID_Service can be written to:

Important: CanWriteCharacteristic must use UUID_CMD_Characteristic, not the temperature characteristic.

2) ML4K prepare (einmal beim Start)

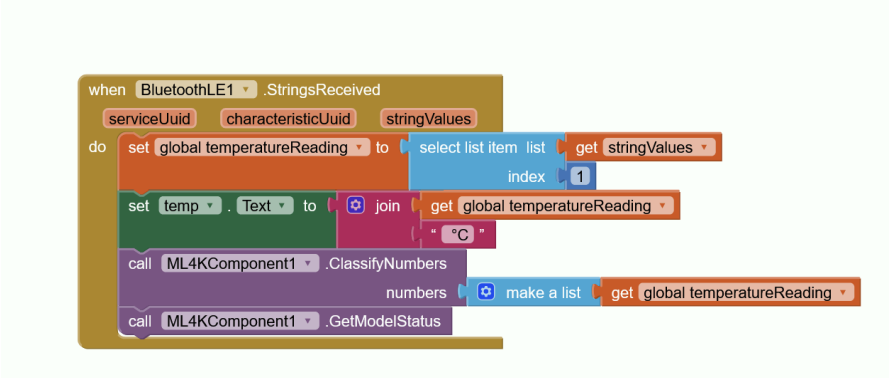
- `ML4K1.Key set`



3) temperature receive & an ML schicken

- In stringsReceived(value):

    1. temperature display
    2. **number merken (temperatureReading = number(value))**
    3. **ML4K aufrufen: ClassifyNumbers(make a list(temperatureReading))**



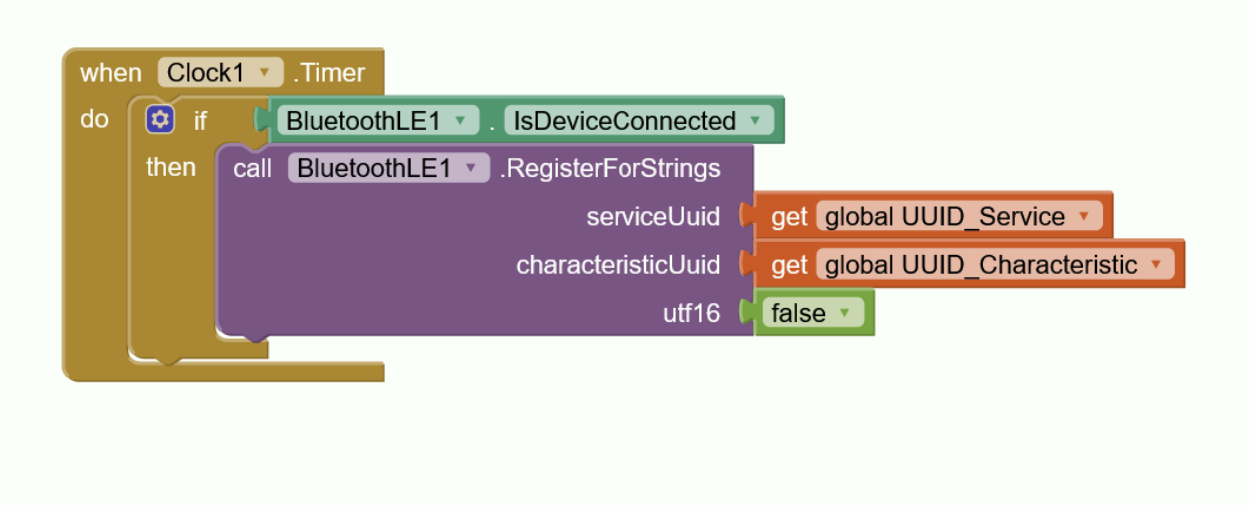
4) ML-Ergebnis → Short send

- In when ML4K1.GotClassification(label, confidence):
    - Mapping festlegen:
        - `label = "hot" → cmd = 1`
        - `label = "cool" → cmd = 2`

        - otherwise → cmd = 3
    - Optional: Only send when the state has changed (if cmd ≠ prev_cmd then … set prev_cmd = cmd).

5) Re-Register & Robustheit

- Mit Clock1 regularly (e.g. all/every 4–5 s) registrieren, solange connected (Wie in AB3.2):
  RegisterForstrings(UUID_Service, UUID_TempCharacteristic, utf16=false)



Test-Protokoll (with Arduino)

1. Connect → status “Connected“.
2. temperature kommt all/every ~5 s (label aktualisiert).
3. **Hand on sensor → ML label = “hot” → short 1 is sent → Arduino: red + fog on.**
4. Cooler place → “cool” → short 2 → green, fog off.
5. In between → “normal” → short 3 → yellow, fog off.

Troubleshooting

- **No sending: canWrite wirklich true? Richtige Characteristic?**
- **No ML result: Is ML4K1.Key set? Is the Numbers project trained/published? Was ClassifyNumbers(list(number)) called?**

- **No temperature: RegisterForstrings(..., utf16=false); device nah genug (≤ 3 m).**
- **(Optional) Note an on/off buffer in the app or on the Arduino.**
- **Need bytes instead of shorts? Dann WriteBytes(...,[1/2/3]) verwenden.**

Presentation notes (2–3 bullet points)

- “If/Else wurde today durch ML4K-Numbers ersetzt (label → Short).“
- “Mapping: hot=1 / cool=2 / normal=3; Arduino schaltet traffic light+fog.“
- “Stability: re-register via Clock, canWrite check before sending.”